BIT-LET

SZUPER

AZ ÖTLET SZÁMÍTÁSTECHNIKAI KÜLÖNKIADÁSA



Grafika: PRIBELSZKY PÁL

Ára: 65,– Ft

SZUPER BIT-LET

A BIT-LET első száma 1983 októberében jelent meg. A legutóbbi pedig 1986 februárjában. A kettő között 28 szám. Ez a Szuper BIT-LET a 28 szám közül elsősorban az első 12 legjobb anyagait tartalmazza. De sok minden bekerült a későbbi, az elmúlt évben megjelent számokból is. Szuper vagy nem szuper, BIT-LET-ünk alapjai nem változtak a kezdet óta. Így hát nyugodt szívvel vállaljuk az akkori Toborzót, s bízunk abban, hogy ma is akadnak, akik elolvasván, sorainkba állnak!

Sajnálatos módon szerkesztőségünk számítógépe egyelőre még nem képes zeneszerzésre, így nem sikerült megfelelő toborozó verbunkost komponáltatnunk az ünnepi alkalomra (persze ami késik, nem múlik!). Így hát csak szavakban fordulhatunk a nagyérdemű publikumhoz, a tisztelt regrutajelöltekhez:

Ne kíméljenek bennünket!

Árasszák el szerkesztőségünket minél több információval, kritikus és behízelgő leveleikkel, észrevételeikkel! Célunk a hazai személyi számítógépes társadalom összefogása és megosztása! Összefogás a személyi számítógépek minden rendű és rangú rajongójával, a gépek terjesztéséért, a programokért, a felhasználókért és az egyre több kilobyte-ért. Összefogás a szűklátókörűség, kisszerűség, a személyi számítógép terjesztését akadályozó rendeletek és téveszmék, valamint a számítógépek terjedésének vámszedői ellen.

Megosztás?

Megosztás a szónak jó értelmében. Nem akarunk semmiféle konfliktust megakadályozni és elhallgatni. Hadd csapjanak össze érvek és ellenérvek. Gyártók és felhasználók, szoftveresek és hardveresek, táborok és ellentáborok, egy-egy gép hívei és ellenfelei. Nem célunk megakadályozni, hogy összevesszenek! Csak egyet kérünk tábornokoktól és közkatonáktól, tisztektől és tiszthelyettesektől! Kard kard ellen, érv érv ellen, szemtől szembe csatázzunk!

Összefogás!

Összefogást hirdetünk a szónak abban az értelmében is, hogy a BIT-LET ne maradjon szerkesztőségünk, valamint már az első szám megjelenésében segédkező számítógépes barátaink magánügye. Szeretnénk, ha minél több szakértő és érdeklődő lenne nemcsak olvasónk, hanem aktívan közreműködő munkatársunk is! Toborzói hagyományok szerint ilyenkor illik elegendő zsoldot, bort, búzát és békességet, valamint fűt és fát ígérni. Mi fű és fa helyett Gépfrontot hirdetünk!

Előre a Gépfrontért!

Kezdők és haladók, géppel rendelkezők és gépért áhítozók, oktatók és okítottak, hardveresek és szoftveresek, műszakiak és felhasználók, profik és amatőrök egyaránt találjanak nekik szóló olvasnivalót! Jelszavunk még ezen kívül:

Arccal a suli felé!

Arccal azok felé, akik ma tanulják a gépkezelés és géphasználás alapjait! Hiszen rajtuk múlik, hogy lesz-e a felhasználásra kellőképpen felkészített, kellőképpen megfertőzött nemzedékünk. Az összefogás sikere rajtuk is múlik, az összefogás eredményessége rajtuk keresztül, rajtunk áll vagy bukik!

Toborozzuk híveink közé mindazokat, akik jelszavainkkal és célkitűzéseinkkel egyetértettek, akik képesek gondolatainkat elolvasni, saját egyetértésüket vagy egyet nem értésüket pedig akár botladozón is mondatba foglalni! És végül, de nem utolsósorban kérjük, hogy ne feledjék:

Lapjaink meg vannak számlálva!

Tizenhat oldalra vagy húszra csak tizenhat-húsz oldalnyi közlendő fér. Havonta egyszer, vagy esetleg majd havonta kétszer ennyit vagyunk képesek Önökhöz eljuttatni. De nem felejtünk! Egyetlen közlésre érdemes gondolatfoszlányt sem engedünk át az enyészet martalékának!

Ne feledjék: A szerkesztő azért van, hogy a lap olyan legyen, mint amilyenek az olvasói!



VALLATÓ 2–49. o.

A BIT-LET eddigi történetének egyik legsikeresebb rovata lett a VALLATÓ. Fölidézzük valamennyi gép vallatását, tekintet nélkül arra, hogy az idő bennünket igazolt vagy sem. Így fordulhat elő, hogy jóslásaink vagy jövőbe vetített óhajaink és sóhajaink nem mindig jöttek be!



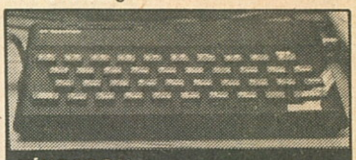
KÍNPADON A ZX 81 2–5. o.

„A gép forradalmat jelentett. Mind műszaki megoldásait, mind az árát tekintve a fantasztikum határát súrolta megjelenésekor. Amióta kiforrottabb konstrukciók is piacon vannak, komoly számítógépes nem vásárol ZX 81-et, de változatlanul ajánlható minden kezdőnek" – írtuk két évvel ezelőtt. S írjuk és valljuk ma is!



KÍNPADON A HT 1080Z 6–11. o.

„...Az inkvizítorok átlagban pontosan négyes osztályzatot adtak a szubjektív véleményre. Szerkesztőségünket egy kicsit meglepte ez a magas átlagosztályzat, de úgy látszik, mi hallottunk eddig sok rosszat a HT-ről, pontosabb vizsgálat esetén kiderül, nincs is olyan sok baj vele." – Az a baj, hogy az osztályzatok, s a Vallatás óta szerzett személyes tapasztalataink is arra intenek bennünket – ez a vallatás egy kicsit rózsaszínűbb lett, mint a valóság!



KÍNPADON A ZX SPECTRUM 12–15. o.

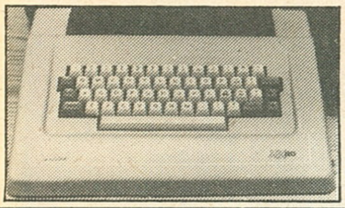
Egyik inkvizítorunk írta a Spectrumról: „A számítástechnika népszerűsítését egyedülállóan szolgáló »cégecske« szinte diktálja a feltételeket a nagyoknak. Olcsóságát nem az eladhatatlan készletek, hanem a zseniális műszaki megoldások eredményezik. A ZX 80–81 gyerekbetegségeit nagyrészt orvosolták, és olyan játékszert bocsátottak ki, amit felnőtté válva sem fog eldobni tulajdonosa, inkább megszállottként védelmezi majd." Nos az utolsó mondat kiállta az idő próbáját, de ami az eladhatatlan készleteket illeti ...?

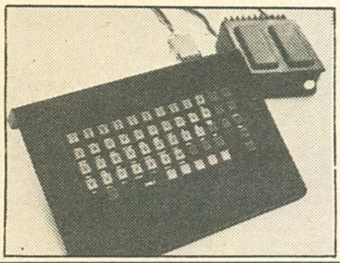
KÍNPADON A COMMODORE 64 16–19. o.

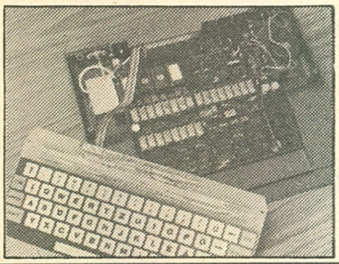
„A gyártók játékgépnek készítették, ám hazai magas ára egyelőre lehetetlenné teszi, hogy bárki ezért vásárolja meg. A nagy gépekhez viszonyított alacsony ára azonban indokolttá teszi, hogy más

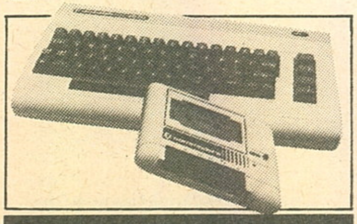
BELÜLRŐL





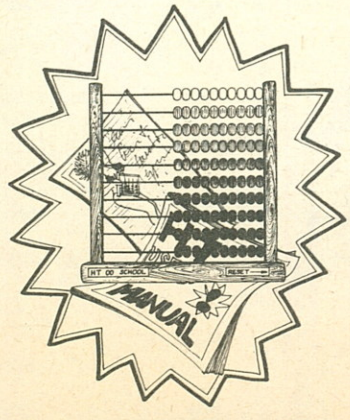








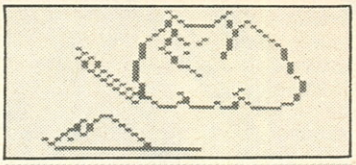




CSM CSINÁLD MAGAD

PROGRAMOK – ÖTLETEK

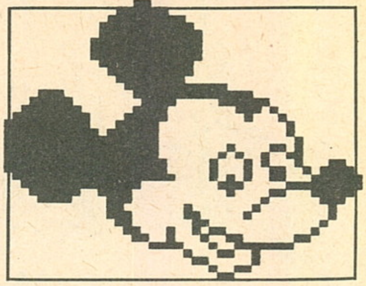
ZX 81



A szerkesztő azért van, hogy a lap olyan legyen, amilyenek az olvasói!

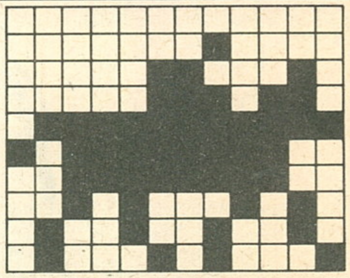
ZX SPECTRUM

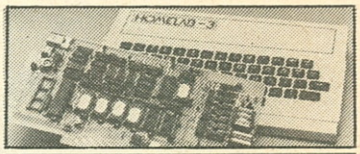
HT 1080 Z



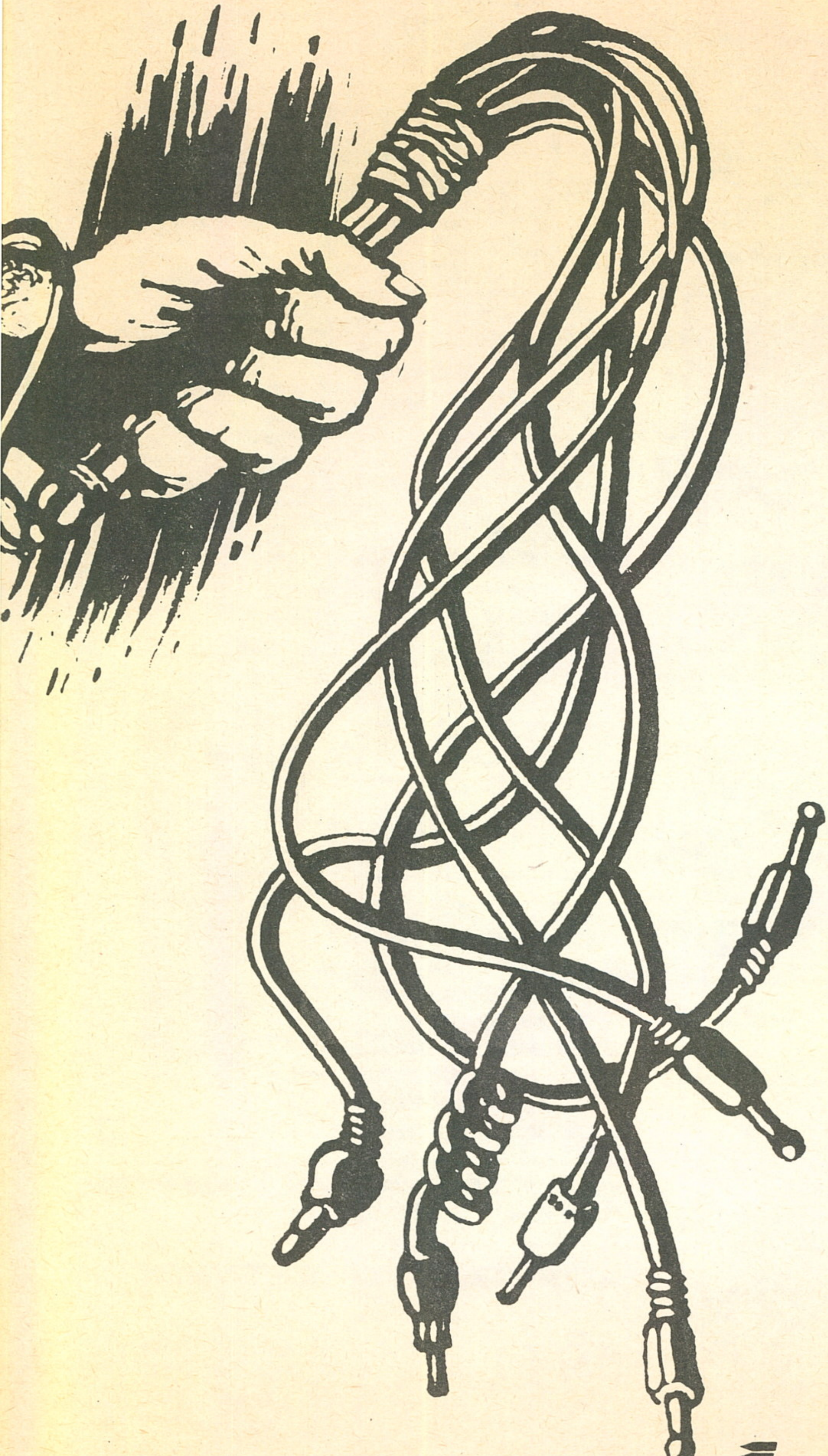
ABC 80

PRIMO





Szuper BIT-LET az ÖTLET számítástechnikai különkiadása + Felelős szerkesztő: Erdős Ákos + Szerkesztő: Angyalosi László + Szerkesztőség: Budapest XIII., Jász utca 103–105. + Levélcím: Budapest 1986 + Telefon: 403-743, 403-797 + Kiadja az Ifjúsági Lap- és Könyvkiadó Vállalat + Felelős kiadó: dr. Petrus György + Készült a Zrínyi Nyomdában Budapesten 1986-ban + Terjeszti a Magyar Posta



VALLATÓ

Két igazán egyszerű szerszám, még két kalapács is különbözhet egymástól: az egyiknek simább a nyele, jobban simul a kézhez, a másik viszont egyenesebb fejű, ritkábban csúszik le a szögről! Hát még két számítógép! Az egyik okosabb, a másik butább, ez jobban hallgat a parancsra, az meg egyszerűen oldja meg a feladatot. És folytathatnánk. Vallató rovatunkban egy-egy személyi számítógép vizsgálatát végezzük el, sorra véve a Magyarországon már elterjedtebbnek tekinthető típusokat. Vállaljuk, hogy szubjektívek leszünk! Nem akarunk gyári adatokkal vitába szállni, éppen az a célunk, hogy a viszonylag hasonló kategóriájú gépek közötti különbségeket keressük, éppen azokat az eltéréseket, amiket a prospektusok nem tartalmaznak, amit csak a felhasználók ismernek. Ezért mindig olyan vallatókat keresünk, akik az éppen kínpadon levő gépet jól ismerik, gyakran használták már.
A ZX 81 méltó arra, hogy elsőként állja végig a kínokat, még akkor is, ha az adott osztályzatokból kiderül, hogy nem vizsgázott kitűnőre.

GYÁRI ADATOK
Memóriaméret: 1 kbyte beépítve, tartozékként kapható 16 és 64 kbyte.
Méret: 167x175x40 mm
Súly: 350 gramm
Billentyűzet: igen egyszerű, fóliaérintkező megoldás.
Beépített magnetofon: nincs.
Perifériák: tv-készülék (csak fekete-fehér), kazettás magnetofon, printer, I/O interface, beszédszintetizátor.

KÍNRENDSZER
A ZX 81 vallatása közben dolgoztuk ki azt a kínrendszert, amely a többi személyi számítógépre is alkalmazható. A közösen elfogadott 13+1-es kínrendszer alapján alakult ki a táblázatunk, amely a vallatók által adott osztályzatokat mutatja, és ezek átlagát. Nem osztályzott mindenki minden szempontot, csak azokat, amelyekben járatos, és amelyekben úgy ítélte, hogy egy osztályzattal kifejezheti a véleményét. A táblázatból látható, hogy az iskolai módszert választottuk egytől ötig. A kapott átlagok így is érdekesek, azonban – hogy a legszélsőségesebb vélemények is érthetőek legyenek – szükségesnek éreztük az egyes kínok magyarázatát.

Ár – 4,2

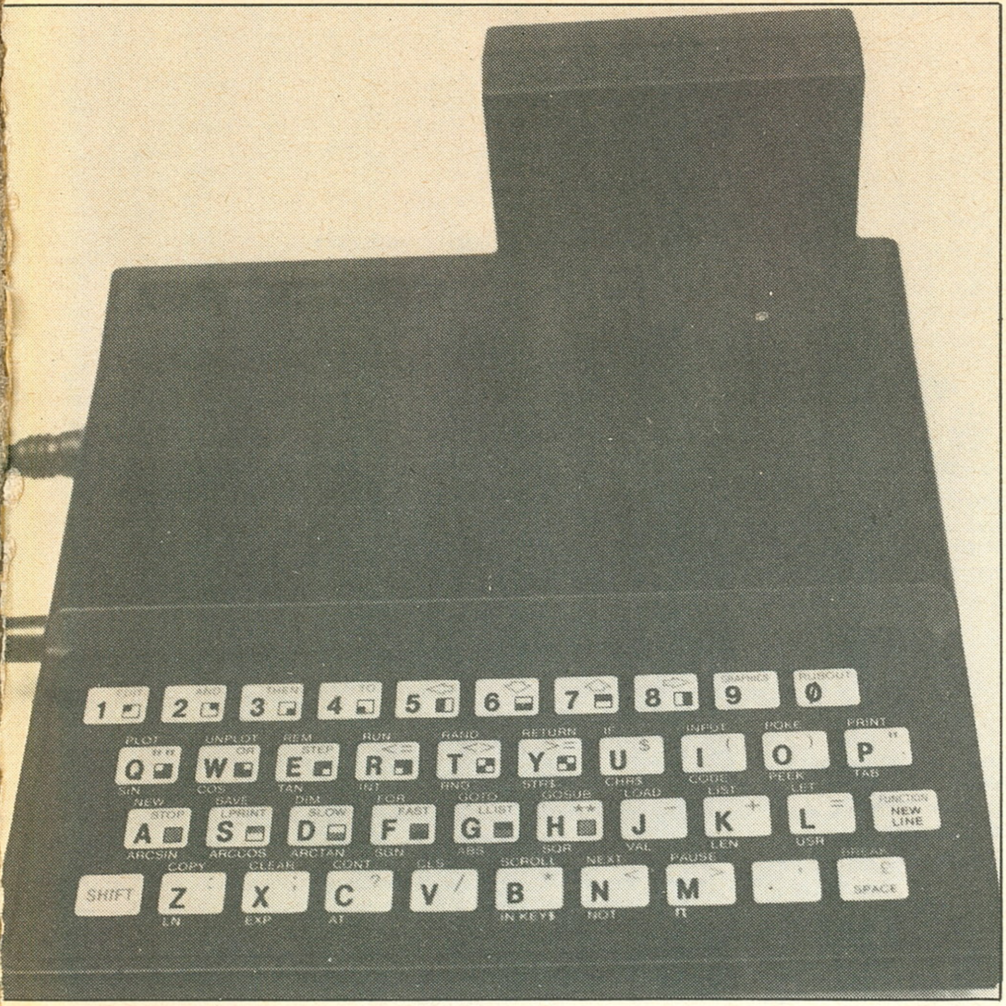
1 KÍN

Ez bizony nagy kín! Elsősorban itthon. Olyannyira, hogy megbukott, osztályozhatatlan. Tény, hogy a többi géphez viszonyítva még mindig a legolcsóbb, azonban az árnál is, és egyéb szempontjainknál is a gépeket igyekeztünk saját magához viszonyítani, és nem egy elvont, nem létező ideálhoz vagy a meglevő egyéb típusokhoz. Az ár kérdését tehát a vallatók így tették fel: megéri-e a pénzét? Itthon nem, külföldön igen. Zoletnik Sándor így fogalmazott: „Ára megjelenéskor forradalmian alacsony volt, azóta rohamosan csökkent. Lassan közeledik a hamburger árához".

Perifériák – 3,1

2 KÍN

Ebben a pontban mindig azt osztályozzuk, hogy az alapgép milyen kiegészítő berendezésekkel kapcsolható össze. Hosszan kerestük az elnevezést a „bővíthetőségtől" a „kapcsolatokig". Ez utóbbit azzal vetettük el, hogy túlzottan



érzelmes. Végül kemény munkával megtaláltuk a már évek óta használt kifejezést: a perifériákat; itt azonban azok minőségét nem osztályozzuk, kizárólag azt, hogy milyen és mennyi periférialehetőség van. A ZX 81-nél megoszlanak a vélemények, aminek egyik oka, hogy Magyarországon hozzáférhető periféria kevés van, és az is drága. Ebbe a pontba tartozik a memóriabővítési lehetőség is, amelyre ennek a gépnek nagyon is szüksége van, hiszen a beépített 1 kbyte inkább „felejtés"-nek nevezhető, mint „memóriá"-nak. A ZX 81 könnyen bővíthető 16, illetve 64 kbyte-ra, de alkalmanként az érintkezési hibák miatt mindent elfelejt.

Képernyőkezelés – 2,7

3 KÍN

Beletartozik a színek lehetősége, a felbontóképesség, a tárolható oldalak száma, a feliratozási lehetőség, egyszóval mindaz, ami a képernyőn megjelenik. A ZX 81 csak fekete-fehér képet tud, a kép előállításának módja igen egyszerű, forradalmi konstrukció. Nagy hibája, hogy a képelőállítás befolyásolja a gép sebességét: ha a képernyőt kikapcsoljuk, lényegesen gyorsabbá válik. Kép nélkül viszont nehéz az élet! Felbontóképessége gyenge, 44x64 pixel. Ékezetes betűk nincsenek, összesen 22 sorba, soronként 32 karakter írható. A képernyő pontonkénti tartalmát nem tárolja a gép.

Hang – 1,6

4 KÍN

A legjobb indulattal sem állíthatnánk, hogy a ZX 81 a muzsika klasszikusain nevelkedett. Közvetlen BASIC utasítással előállítható hangja nincs, de zenei hangját „előhívhatjuk" gépi kóddal. Érdemes?

ZX 81
VALLATASANAK EREDMENYE

K I N O K	REVAI VERA KOZEPISK. TANULO	BANYI LASZLO FOISK. HALLGATO	SZENTTORNYAI LASZLO FOISK. HALLGATO	ZOLETNIK SANDOR EGYETEMI HALLGATO	KEPES JANOS MATEMATIKUS	SZEKELY JENO FOISK. DOCENS	HALASZ PETER EGYETEMI HALLGATO	DR. TOROK TURUL MATEMATIKUS	ATLAG
1 .KIN:AR	-	4	4	-	5	5	3	4	4.2
2 .KIN:PERIFERIAK	5	2	3	3/4	3	-	3	2/3	3.1
3 .KIN:KEPERNYOKEZELES	3	2	2	3	4	3	3	2	2.7
4 .KIN:HANG	-	1/2	1	-	2	2	1	2	1.6
5 .KIN:KAZETTAS TAROLAS	3	2	2	-	3	3	2	2	2.4
6 .KIN:GEPI KODU PROGRAMOZAS	-	3/4	4	2	3	3	2	2	2.8
7 .KIN:MEGBIZHATOSAG	3	3	1	3	3	2/3	2	-	2.5
8 .KIN:BILLENTYUZET	3	2	1	2	3	2	2	2	2.1
9 .KIN:DOKUMENTACIO	4	4	5	-	5	5	4	3	4.3
10 .KIN:EDITALAS	5	3	3	4	5	4	3	3	3.7
11 .KIN:A GEP PROGRAMNYELVE	4	2	3	-	4	-	3	3/4	3.2
12 .KIN:TANULHATOSAG	-	4	5	-	5	-	4/5	4/5	4.6
13 .KIN:EMBERKOZELSEG	-	3	4	-	5	-	4	4	4.0
+ 1 KIN:SZUBJEKTIV VELEMENY	4	-	-	-	5	4	-	-	-
ATLAG	3.8	2.8	2.9	2.9	3.9	3.3	2.8	2.9	3.2

Tévedni emberi dolog, de igazán összekutyulni valamit csak számitógéppel lehet!

SZUPER VALLATÓ

Kazettás tárolás megbízhatósága – 2,4

5 KÍN

Beépített magnetofonnal nem rendelkezik, viszont kapcsolható hozzá bármilyen típus. Rutinos ZX-esek szerint kellő gyakorlással kikísérletezhető megbízható beolvasás; ezzel sokan próbálkoznak, de keveseknek sikerül. A beépített szintszabályozó automatika nem kielégítő. Tény, hogy vallatásunk szünetében tettünk kísérletet néhány egyszerű játék beolvasására, amely harmadszorra sikerült is.

Gépi kódú programozás lehetősége – 2,9

6 KÍN

Monitor üzemmód nincs, gépi kódú programozás lehetséges, egyesek szerint túl bonyolult, mások szerint igen egyszerű, de szinte mindenki úgy ítéli, hogy egy ekkora gépnek éppen elegendő.

Megbízhatóság – 2,5

7 KÍN

Egységes a vélemény: a csatlakozók állandó hibalehetőséget jelentenek és gyakori a gép túlmelegedése.

Billentyűzet – 2,1

8 KÍN

A gép olcsósága elsősorban a billentyűzet egyszerűségéből adódik, amely technikailag forradalmi újdonságokat is hozott. Ennek ellenére vannak, akik meg tudják szokni. Általános vélemény az, hogy gyermekeknek megfelelő, a felnőttek kézméretük és megszokásaik miatt azonban idegenkednek tőle.

Dokumentáció – 4,3

9 KÍN

Ebben a pontban kizárólag a gép vásárlásakor kapott írott anyagról van szó, tehát nem minősítjük az egyébként fellelhető, szakirodalomban megtalálható dokumentációt. Érdekes vita alakult ki ezzel kapcsolatban: akik nem ZX 81-en tanultak, úgy ítélték meg: gépkönyvnek jó, BASIC tankönyvnek azonban használhatatlan a dokumentáció. Aki viszont ebből tanult, az tankönyvként is kitűnőnek értékelte.

Editálás – 3,7

10 KÍN

Ebbe a pontba tartozik a szerkesztés, az utasítások módosítása, a szintaktikai ellenőrzés és javítás is. Általános vélemény, hogy az editálási lehetőségek kielégítőek, több szempontból kiválóak: karakterek beszúrására is van lehetőség. Szemléletes, ahogyan az elfogadott programsorokat felviszi a képernyő tetejére.

A gép programnyelve – 3,3

11 KÍN

Általános vélemény, hogy a gép programnyelve, a ZX 81 BASIC kicsit szűk, de kezdőknek jól használható utasításkészlet. Az osztályzatok között nagy eltérés is van, mert a hiányzó utasításokat szellemesen megkerülők jobbnak értékelik a gép nyelvét, mint a lustább programozók.

Tanulhatóság – 4,6

12 KÍN

Itt nem a nyelvre gondoltunk, hanem a gép kezelésének tanulhatóságára: mennyire speciális, mennyire könnyedén használható. Az alapvélemény ez: könnyen tanulható, de őrjítő. A frappáns vélemény azzal magyarázható, hogy a gép sokfajta újítást tartalmaz, ezért kezdőknek könnyen megtanulható, de aki már más géphez szokott, az gyakran mellényúl. Szélsőségesen megoszlottak a vélemények a BASIC alapszavak egyetlen „gombnyomással" történő előhívhatóságával kapcsolatban.

Emberközelség – 4,0

13 KÍN

Ennek a pontnak a legszebb megfogalmazása nem tőlünk származik, de vállaljuk: mennyire megy a gép a felhasználó után vagy mennyire kényszerül a felhasználó a gép után menni. Ebből a szempontból a ZX 81 szinte kizárólag jó minősítést kapott, általában barátságos, szolgálatkész gépnek tartják. A lehetséges idegenkedés egyetlen forrása a már említett billentyűzet.

Szubjektív vélemény – 4,3

+1 KÍN

Ebben a pontban egy általános, minden objektivitást nélkülöző megfogalmazást vártunk arra, szereti-e használni a gépet valaki vagy sem. Vallatásunk során ismét összecsaptak a ZX-hívők és kevésbé hívők. A véleményekből az derült ki, hogy akik egyszer megszerették a ZX 81-et, azok szinte szerelmesek lettek bele, és haragosan fújtatnak, ha mások a hibákat sorolják. A hibákat sorolók is elismerik azonban, hogy a gép forradalmat jelentett, mind műszaki megoldásait, mind az árát tekintve a fantasztikum határát súrolta megjelenésekor. Amióta kiforrottabb konstrukciók is piacon vannak, komoly számítógépes nem vásárol ZX 81-et, de változatlanul ajánlható minden kezdőnek.

BRÁNYI LÁSZLÓ: Szegényes karakterkészlet, fekete-fehér képernyő

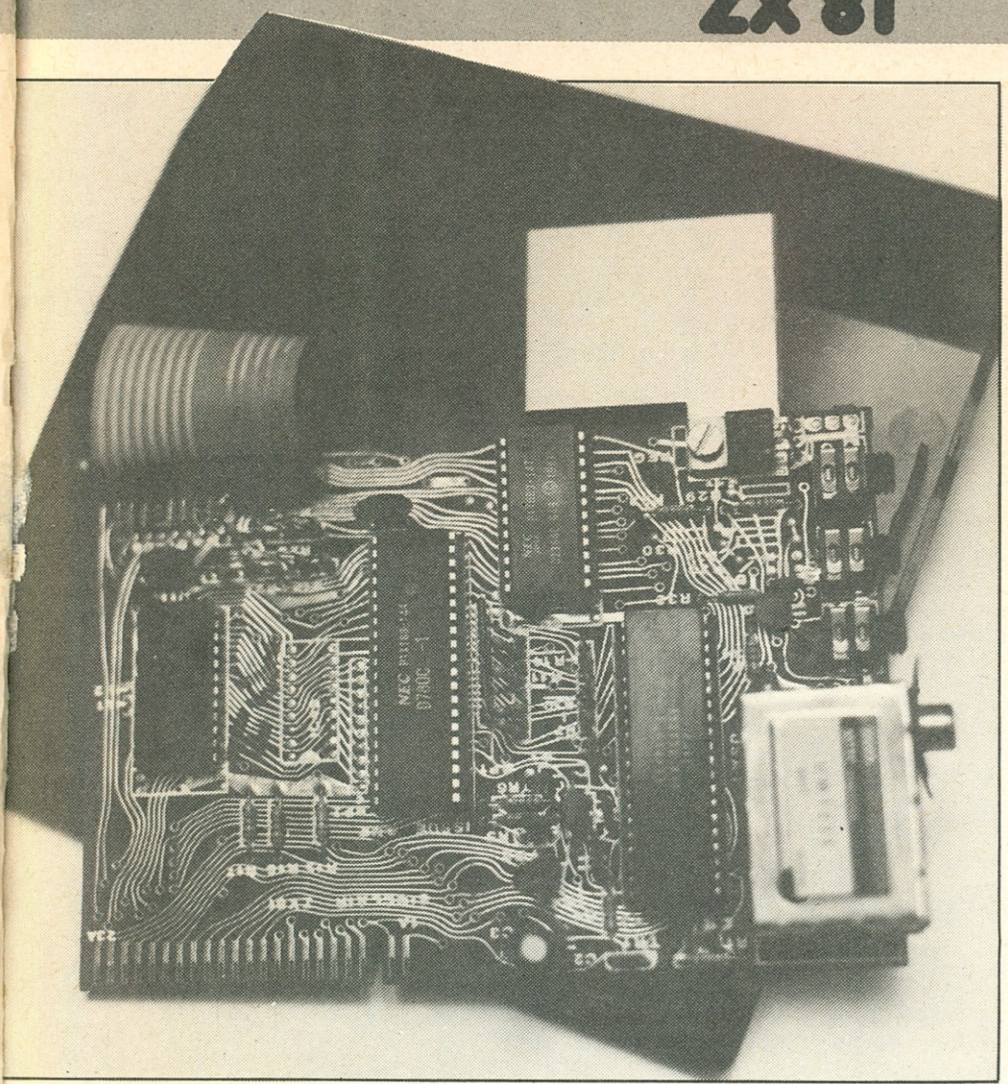
KEPES JÁNOS: Az ára? Laikusoknak, kezdőknek nagyon megéri!

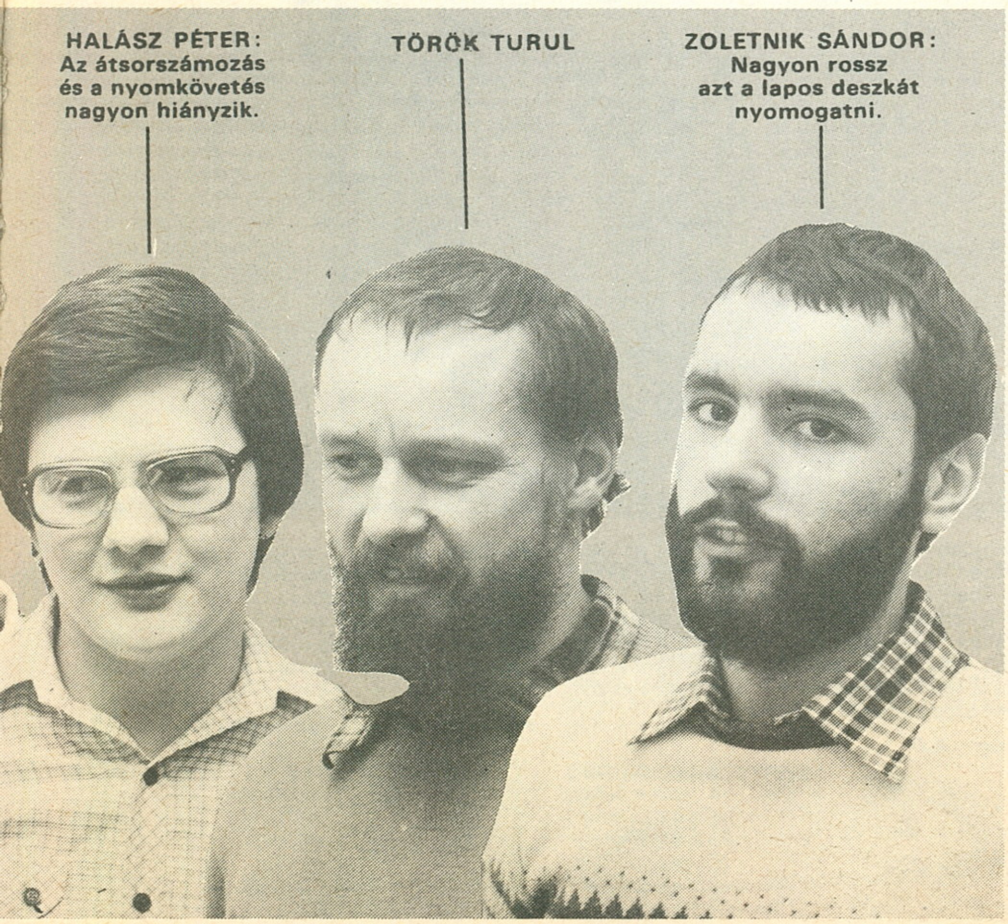
SZÉKELY JENŐ: A dokumentáció: Gépkönyvként jeles, BASIC tankönyvként nem minősítem!

RÉVAI VERA: Többször előfordult, hogy fél óra után a gép hirtelen kikapcsolt és mindent elfelejtett.

SZENTTORNYAI LÁSZLÓ: Nagyon gazdaságos memóriakihasználás!







A VÉDELEM NEVÉBEN

Elolvastam a ZX 81 „kín"-os vallatásáról készült jegyzőkönyvet, és megszületett az első elhatározásom, hogy kéretlenül is a védelem nevében szólaljak meg. A kínrendszerrel nem kívánok vitába szállni, de meg kell jegyeznem, hogy könnyen hamis kép alakulhat ki az olvasóban, és főként a leendő személyi-számítógép-tulajdonosokban, ha a gép szolgáltatásait, teljesítőképességét elvonatkoztatjuk annak árától.

Egy Trabant vagy Lada jellemzőiben lényegesen eltér, és nem mindig hasonlítható össze például egy Mercedesszel, de a közlekedési szabályok mindegyik típusra azonosak, és mindegyik alkalmas eszköz egy adott cél elérésére. Viszont ma Magyarországon a forgalomban részt vevők közlekedési kultúráltságát döntően a Trabant stb. vezetőkön lehet lemérni, függetlenül attól, hogy ők is szívesen vezetnének Mercedest. Márpedig ma, és valószínűleg a közeljövőben Magyarországon az általános számítástechnikai műveltség Trabantja a ZX 81, melyen a „vezetési gyakorlat" legalább olyan jól (ha nem könnyebben) elsajátítható, mint társain. A ZX 81 azonban rendelkezik egy olyan előnnyel, melyet a többiek nem mondhatnak el magukról, nevezetesen: ez igazodik legjobban a vevő pénztárcájához, és ez nem is lényegtelen szempont.

A gép vallatási jegyzőkönyve bizonyára annak legegyszerűbb változatain szerzett tapasztalatokat tükrözi. Tekintettel arra, hogy ELKON gazdasági munkaközösségünk ezzel – is – foglalkozik, hadd hívjam föl az olvasók figyelmét arra, hogy ZX 81 továbbfejleszthető hető! Íme néhány lehetőség, amelyek megvalósulásában mi is partnerei lehetünk a ZX-tulajdonosoknak.

Perifériák

Létezik már hazai gyártású, 16 kbyte-os memóriabővítés, melynek azonban igazi szépséghibája, hogy a szocialista és hazai gyártmányú alkatrészek ára miatt lényegesen nem olcsóbb az „import" memóriáknál. Előnyének mondható, hogy a csatlakoztatás érintkezési hibája ennél nem jelentkezik. Lényegesnek mondható periféria lenne a nyomtató, melynek ára nyugaton is többszöröse a gép árának. Ha valaki mégis rendelkezik valamilyen mátrixnyomtatóval, annak lehetősége van azt csatlakoztatni a ZX 81-hez hazai gyártású bővítésen keresztül is.

Képernyőkezelés

Nem boszorkányság, de a gép felhasználója tetszőleges 8x8 pontból álló karaktereket definiálhat az eredeti karakterek helyett. Ehhez csupán vagy a gép hardver átalakítása, vagy pedig bővítése szükséges, mely talán már nem is olyan soká a magyarországi ZX 81 tulajdonosok számára egyszerűen megszerezhető lesz.

Kazettás tárolású megbízhatóság

A túlmelegedés kiküszöbölhető, a beolvasási biztonság pedig nagymértékben növelhető az alapgépen végrehajtott hardvermódosítással.

Amiről nem volt szó

A kezelés kényelmét növelik az alapgépen történő további módosítások, mint pl. a RESET gomb, tápkijelzés, hangjelző. Az utóbbi minden elfogadott billentyűmegnyomás után csipogó hangjelzést ad. Elsősorban játékprogramoknál jól használható a cursor mozgatására a botkormány (joystick), mellyel a négyirányú mozgatás, plusz egy vezérlő jel kiadása lehetséges. Talán ez sem marad sokáig álom. **Kovács G.**

A HT 1080Z Schoolcomputer (továbbiakban HT) – azzal, hogy ma már szinte az összes középiskolában megtalálható – nagyobb jelentőségű bármelyik típusnál, hiszen a következő nemzedék éppen ezzel a géppel találkozik majd elsőként. Hogy megszereti-e a HT-t, vagy elmegy a kedve a számítástechnikától – vallatónkból kiderül.



GYÁRI ADATOK

Csatlakozási lehetőségek: második magnetofon, output port, printer, floppy drive, memóriabővítés (utóbbiak perifériacsatolóval).

Méret: 540x390x120 mm.

Súly: 3 kg.

Memóriaméret: 12 kB (ROM), 1,5 kB (monitor), 16 kB (RAM).

Beépített magnetofon: 4,75 cm/sec szalagsebességű, compact kazettás, fémmechanikás.

KÍNRENDSZER

A ZX 81-es vallatása közben kidolgozott kínrendszert használtuk egy apró, de fontos bővítéssel: amiről külföldi gépeknél még nem beszélhetünk, arról a HT-nál érdemes, a szervizlehetőségről. Így a 13+1-es kínrendszerünk a kissé slamposabb 13+2-esre módosult, de talán megérte. Táblázatunk sajátossága az is, hogy a vallatásra meghívtuk a gyártó cég képviselőjét is – nem utolsósorban azért, hogy a fejlesztésről is információkat kapjunk. Az általa adott osztályzatokat közöljük, azonban a tisztességes játék érdekében az átlagokba nem számítottuk bele. (Annak ellenére, hogy Tóth Ferenc osztályzatai olykor szigorúbbak a többiekénél.)

Ár – 3,6

1 KÍN

A gép megjelenése óta valamelyest csökkent az ára, szinte valamennyi inkvizítorunk lényegében elfogadhatónak tartja, de mindenki hozzáteszi: lehetne olcsóbb is. Ha olyan gyorsasággal csökkenne az ára, ahogy a világpiacon a többi típusé, jövőre már magánembereknek is megérné. De azért senkinek nem ajánljuk, hogy erre várjon!

Perifériák – 3,4

2 KÍN

A táblázatból látható, hogy komoly eltérés volt a vélemények között. Nem csoda, mert ebben a kínban azt osztályozzuk, hogy milyen a perifériák csatlakoztatási lehetősége. Tény, hogy a HT-hez szinte minden kapcsolható adapteren keresztül, csak az a bökkenő, hogy nincs mit és nincs min keresztül. Így aki a „csatlakozási lehetőséget” úgy értelmezte, hogy van-e lyuk, ahova be lehet dugni valamit, az jelest adott, aki a bedugnivalót is hiányolja, az elégségest. Feltétlen megemlítendő a beépített magnetofon, amely a többség szerint előnyös (ha megbízhatóan működne, valószínűleg még előnyösebb lenne!), és a második, külső magnetofoncsatlakozási lehetőség, amely kitűnő.

Képernyőkezelés – 3,2

3 KÍN

A gép fekete-fehérben dolgozik, kétféle kijelzési formátummal: az elsőben 64 karaktert ír soronként, a másodikban 32-t, de így csak a képernyő felét mutatja. ~~Erről a módszerről~~ nem a legjobb vélemény alakult ki, mert az első esetben nehezen olvashatóak a betűk, a másodikban viszont nehézkes a lapozás. Egyik inkvizítorunk ezt írta erről: „64 olvashatatlan karakter egy kicsit sok, 32 olvasható egy kicsit kevés”. A grafikus felbontóképesség durva: 48x128 képpont, így komolyabb rajzok létrehozására nincs lehetőség.

Hang – 4,6

4 KÍN

Egy mondatban: szépen énekel, ha szóra lehet bírni. A beépített hanggenerátor három csatornán, programozható frekvenciájú négyszögjelet állít elő, ezen kívül háromcsatornás zajgenerátor is van. A hangja szép, programozása azonban meglehetősen bonyolult és suta. Minthogy azonban inkvizítoraink „profi” programozók, s így ez nekik nem okoz gondot, ezért a kitűnően működő hangrendszert igen magasra értékelték. Kezdők azonban óvakodjanak!

Kazettás tárolás megbízhatósága – 3,6

5 KÍN

Ahhoz képest, hogy a kapott átlagosztályzat egyáltalán nem rossz, meglepő, hogy szinte mindenki ezt írta a papírjára: megbízhatatlan. Az ellentmondás úgy oldható fel, ha a kínban foglaltakat értelmezzük. A „kazettás tárolás

megbízhatósága" elnevezésbe tartozik a program beolvasása, az adatok be- és kiolvasása, a második magnetofon működtetése, a program kimentése és ellenőrzése. Inkvizítoraink általában a sokféle funkció közül eggyel elégedetlenek, azért írják, hogy megbízhatatlan, viszont az összes többivel elégedettek, ezért adtak jó osztályzatot. A szép az lenne, ha most leírhatnánk, melyikkel elégedetlenek, azonban sajnos lehetetlen. Ugyanis mindenki mással elégedetlen! Lehet, hogy a hiba nem is a gépben van? Az mindenesetre tény, hogy a beépített magnetofon mechanikája igen gyenge. Ezt a gyártók is elismerik azzal, hogy a fejlesztett típus már új, megbízhatóbb magnetofont tartalmaz. Egy a szűkszavú vélemények közül: „A programot gyakran sikerült betölteni. (Azóta vannak tapasztalataim az új magnóról. Nem tökéletes, de jobb.)"

Gépi kódú programozás lehetősége – 4,5

6 KÍN

A vélemény egyértelmű, van monitor üzemmód is, jól használható. Egyetlen hiba, hogy a gépi kódú programok csak segédprogrammal menthetők ki, fölöslegesen bonyolítva a felhasználó életét.

Megbízhatóság – 4,5

7 KÍN

Külön öröm egy magyar gyártmányú berendezésről azt írni, hogy igen megbízható! Ebben az országban, ahol még a hazai gyártmányú vizespohárból sem ott folyik ki a víz, ahol kellene; komoly eredmény, hogy egy számítógép hosszú időn keresztül hiba nélkül működik. Ezért minden tisztelet a gyártó Híradástechnika Szövetkezeté. Lehet persze, hogy vizespoharat ők sem tudnának készíteni – az egy kicsit bonyolultabb!

Billentyűzet – 3,9

8 KÍN

Az osztályzat jó, az általános vélemény: kézhez álló, gyorsan használható, kellemes, kényelmes. Hibaként megemlítették, hogy spontán betűismétlésre hajlamos. A gyártó tájékoztatása szerint a fejlesztett változat javított billentyűkezlése már megszüntette ezt a hibát is.

Dokumentáció – 2,2

9 KÍN

A kapott osztályzatok közül kimagaslóan a legrosszabb, az egyszavas megfogalmazás ez lehet: csapnivaló. Egyik inkvizítorunk mentségképpen ezt írja: „Nehéz jó tankönyvet és egyben jó gépleírást is adni". Ez igaz, a HT dokumentációjának azonban az a baja, hogy minden funkcióra alkalmatlan. Gépkönyvként túl kevés információt tartalmaz (csak egy példa, hogy cikkünk alapadatait is – pl. memóriaméret, csatlakozási lehetőségek – máshonnan kellett beszereznünk). Tankönyvként viszont zavaros felépítésű, sokszor hibás, kezdőknek érthetetlen, de néha a haladókat is kemény feladat elé állítja. Sajnos a módosított kiadás is csak a sajtóhibák kijavítását vállalta (azt sem teljes sikerrel!), a módszertani átdolgozást nem. Nagyon hasznos lenne, ha egy egyébként jó színvonalú termék gyártója vállalná, hogy ezt a nagyon fontos apróságot is kezelné, fejlesztené.

Editálás – 3,4

10 KÍN

A HT gép sokféle editálási funkciót tud ellátni. Ez kezdő, vagy közepes szinten túl bonyolultnak tűnik, ezért sokan nem is használják azokat. Aki viszont egyszer megszokta, az esküszik rá, hogy kitűnő. Ezért látható a nagy ellentmondás az osztályzatokban. Azt azonban hibának tartotta még a legelszántabb HT-hívő is, hogy a gép nem ellenőrzi szintaktikailag a bevitt programot. Így néha több perces munka vész kárba egy elütés miatt. Másik hiány az, hogy a programszámozás nem javítható.

A gép programnyelve – 4,1

11 KÍN

Nem egyöntetű a vélemény, de a kapott osztályzat így is elég magas. Általában kezdőknek jól használható és tanulható, de a haladók is úgy ítélik, hogy megfelelő. Inkvizítoraink a hiányokat másképpen ítélték meg, abban nem egyeznek a vélemények, hogy mi hiányzik az utasításkészletből – mindenkinek más.

HT-1080 Z

VALLATASANAK EREDMENYE

K I N O K	1 TOTH FERENC VEVDSZOLG.CSOP.VEZ.	2 DR. TOROK TURUL MATEMATIKUS	3 KOVACS MIHALY GIMNAZIUMI TANAR	4 KIRALY ZOLTAN EGYETEMI HALGATO	5 BAN PETER TUDOMANYOS MUNKATARS	6 KISDI BALINT GIMNAZIUMI TANULO	7 KEPES JANOS MATEMATIKUS	8 CZAKO TAMAS KOZEPISK. TANULO	9 TOTH LASZLO MATEMATIKUS	10 HUBERT TIBOR KOZEPISK. TANAR	11 ZSAKO LASZO MATEMATIKUS	ATLAG
1 .KIN:AR	4	4	3	4	4	4	3	4	3	3	4	3.6
2 .KIN:PERIFERIAK	4	3/4	5	2	-	5	-	3	3	2	4	3.5
3 .KIN:KEPERNYOKEZELES	4	4	4	3	4	3	2	3	2	3	3/4	3.2
4 .KIN:HANG	3	4/5	5	5	5	4	4	5	5	4	5	4.5
5 .KIN:KAZETTAS TAROLAS	4	3/4	4	3	3	4/5	3	4	3	4	3	3.5
6 .KIN:GEPI KODU PROGRAMOZAS	4	4/5	5	4	4	5	4	4	4	5	5	4.4
7 .KIN:MEGBIZHATOSAG	-	5	4	4/5	4	5	5	5	4	4	4	4:5
8 .KIN:BILLENTYUZET	4	3	4	4/5	3	5	-	4	3	4	4	3.8
9 .KIN:DOKUMENTACIO	2	2	2	4	2	3	2	2	3	2	2	2.4
10 .KIN:EDITALAS	3	4	4	2	4	4	3	3	1	3	3/4	3.1
11 .KIN:A GEP PROGRAMNYELVE	4	4/5	5	4	3	5	3	3	5	4	4	4.0
12 .KIN:TANULHATOSAG	-	4	4	4	4	4	3	3	2	3	4	3.5
13 .KIN:EMBERKOZELSEG	-	4	4	4	-	5	2	4	3	4	4	3.8
+ 1 KIN:SZUBJEKTIV VELEMENY	4	4/5	5	4	4	5	2/3	4	3	4	4/5	4.0
+ 2 KIN:SZERVIZ	-	5	5	-	5	-	-	-	5	-	5	-
ATLAG	3.6	4.0	4.2	3.7	3.8	4.4	3.0	3.6	3.3	3.5	4.0	3.7

Tanulhatóság – 3,5

12 KÍN

Ennek a gépnek az egyik legfontosabb jellemzője, hiszen a HT alapvetően iskolai oktatási célokat szolgál, tehát laikusok ülnek a gép mellé, ezen ismerkednek a számítógéppel, a programozással. Ha ezt megfelelő súllyal figyeljük, bizony a 3,5 átlag osztályzat nem éppen jó. Általánosságban a gép kezelése könnyen elsajátítható, ezzel tehát nem is lenne probléma. Ami a hibákat illeti, itt a „tanulhatóság" címszóban azok szerepelnek, amelyeket már az előzőekben említettünk. A tanulhatóság rovására megy az olvashatatlan képernyő, a nehezen elérhető hang, a túl bonyolult editálás. Itt érzékelhető a legjobban az, hogy a HT egy licenc alapján gyártott gép, amit Magyarországon oktatási célra használnak. Ez nem mentség, csak tény. Egy érdekes vélemény, ami ide kívánkozik: „Miért vannak angol feliratok a gépen? Egy iskolai számítógépnél ez nem követelmény?"

Emberközelség – 3,8

13 KÍN

Az osztályzat jó, általában úgy ítélte mindenki, hogy a gép hajlandó a felhasználó után menni. Hibaként általában itt is a már tárgyaltak jöttek elő: az olvashatatlan képernyő, a használati utasítás és így tovább. Török Turul így fogalmazott: „A gép eredetije abban a korban született, amikor az emberközelség még nem volt különösebb szempont".

Szubjektív vélemény – 4,0

+1 KÍN

Ezen különösebb magyarázniváló nincsen, mindent egybevetve az inkvizítorok átlagban pontosan négyes osztályzatot adtak. Szerkesztőségünket egy kicsit meglepte ez a magas átlagosztályzat, de úgy látszik, mi hallottunk eddig sok rosszat a HT-ről, pontosabb

vizsgálat esetén kiderül, hogy nincs is olyan sok baj vele. Ez legyen a legrosszabb meglepetésünk!

Szervizszolgálat – 5

+2 KÍN

Fantasztikus győzelem! Veri az összes külföldi gyártmányt, amelyeknek szinte semmilyen szervizlehetőségük nincsen! Erről persze inkvizítoraink közül is csak néhánynak volt információja. Ők azonban csillagos ötöst adtak.

HOZZÁSZÓLÁS

A vallatók közül talán a HT váltotta ki a legnagyobb vihart. Azaz nem is maga a Vallatás, hanem az ahhoz érkezett Simonyi Endre által írott hozzászólás. A hozzászólást, az annak apropójából a Magyar Hírlapban nem sokkal később megjelent cikk részleteit, majd ismét a BIT-LET-ben megjelent a Híradástechnika Szövetkezettől érkezett válaszlevelet elég érdekesnek találtuk ahhoz, hogy beválogassuk összesített vallatónkba.

Megjegyzésként csak annyit:

Simonyi Endre hozzászólása elé bocsátottuk annak idején a következő megjegyzést: „A korrekt tájékoztatáshoz azonban úgy véljük, szükséges közölnünk azt a tényt is, hogy Simonyi Endre Simon '68 nevű mikrogépével maga is érintett volt az iskolaszámítógép-pályázatunkban."

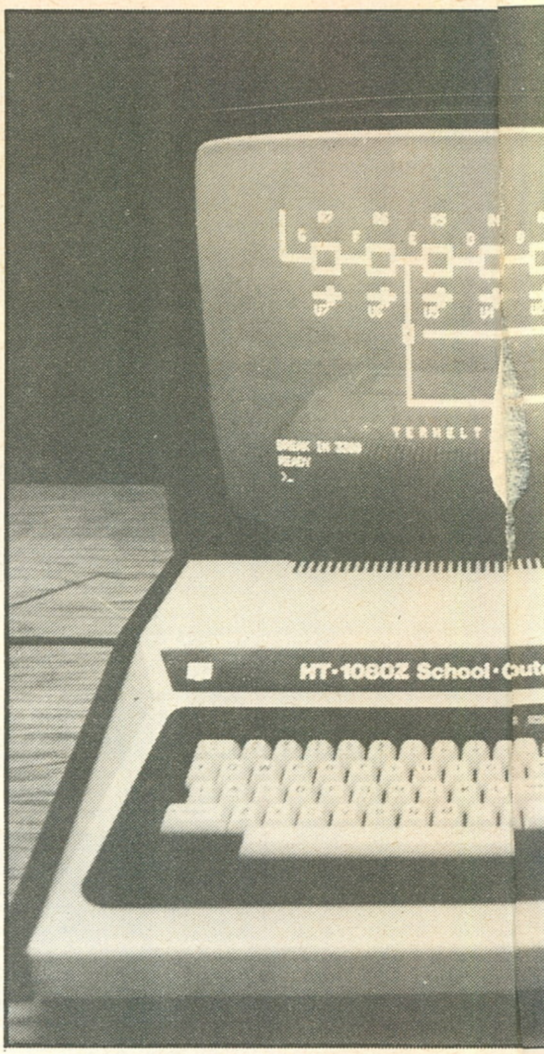
Laptársunk, a Mikromagazin egyik tavalyi számában visszatért az általunk már befejezettnek vélt vitára. Ebből az írásból meglepődve értesültünk, hogy Simonyi Endre nem is vett részt a pályázatban. Meglepődtünk, mert annak idején levele közlésének megbeszélésekor közöltük Simonyi Endrével, hogy lapunk ezt a megjegyzést szükségesnek tartja, s a szerző egyetlen szóval sem tiltakozott, nem említette, hogy ő ne vett volna részt a pályázatban. Így hát továbbra is neki hiszünk, s úgy véljük, részt vett benne.

Laptársunk egyébként azt is szemünkre vetette, hogy a szerkesztőség nem zárta le megnyugtatóan ezt a vitát. Mi úgy gondoljuk, hogy lezártuk. Épp annyival, amennyit szükségesnek tartottunk. Ezt a lezárást most sem hagyjuk el, de eléírjuk, hogy szerkesztőségi lezárás. Talán úgy érthetőbb lesz álláspontunk.

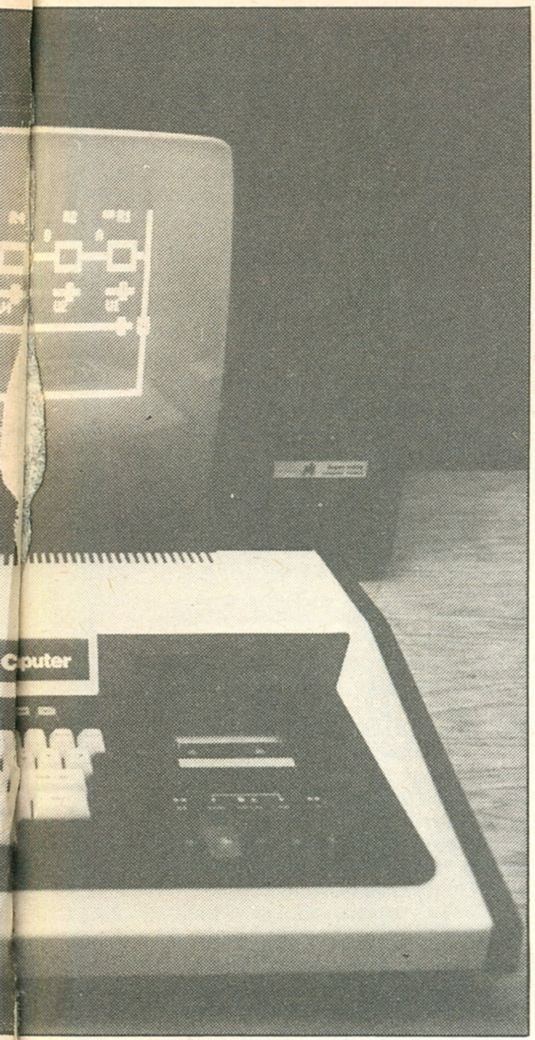
A lap Vallatójában értékelték a HT 1080Z típusú mikroszámítógépet. Az értékelés 7. pontja a gép megbízhatósága volt. Az értékelés összesítéseként közölte: „Külön öröm egy magyar gyártmányú berendezésről azt írni, hogy igen megbízható!... komoly eredmény, hogy egy számítógép hosszú időn keresztül hiba nélkül működik. Ezért minden tisztelet a gyártó Híradástechnika Szövetkezeté."

Ezen dicsérettel szemben áll az, hogy az értékelés 5. kín-ja („kazettás tárolás megbízhatósága") szerint „megbízhatatlan". A gyártó új, megbízhatóbb magnetofont fejleszt – írják. A 8. kín („billentyűzet") szerint „spontán betűismétlésre hajlamos". A gyártó tájékoztatása szerint a fejlesztett változat javított billentyűkezelése már megszünteti ezt a hibát is.

Összegezve tehát az információk gépbe juttatása, kimentése terén megbízhatatlan. Miben megbízható hát? A bejutott információ belső mozgatásában, tárolásában? Ha csak ebben, akkor az egy használhatatlan gép, mert mindenben megbízhatatlan, ami kritikus lehet. Az értékelés összesítése ezen kívül is megdöbbentő eredményt adott, hiszen „jó" minősítést kaphatott egy olyan,







iskolai célra államilag terjesztett gép, ami pl. nem „ismeri" a magyar abc-t („Egy iskolai számítógépnél ez nem követelmény?"– kérdezi az egyik inkvizítor velem együtt), nincs semmiféle megvalósított bővítési lehetőség (bár ez követelménye volt az iskolaszámítógép-pályázatnak), a képernyő megjelenítés igen rossz. Úgynevezett valódi grafika (ami a gyerekek számára a legfőbb vonzerő) nincs, a dokumentációja „csapnivaló" (egy iskolaszámítógépnél ez nem követelmény?, valamint az sem, hogy a magyar szabvány és az iskolaszámítógép-pályázat által kötelezően előírt, és tartalmában meghatározott gépkönyvet a géphez nem adnak stb.)

Ennek ellenére Önök azt hangsúlyozzák, hogy „...ezért minden tisztelet a gyártó Híradástechnika Szövetkezeté". Ezért? Tisztelet?

Egy lapnak tájékoztatni és nem félrevezetni kell az olvasókat. Le kellett volna írni a valóságot – ez a gép iskolaszámítógépnek, így ahogy van, alkalmatlan.

A tájékoztatáshoz az is hozzátartozik, hogy ez a gép (eredetileg VIDEOGENIE 1 nevű távol-keleti gyártmány) túlságosan drága (hiszen a hazaival azonos kiépítésbeni ára 500 DM körüli.) Teljesen elavult típus (hiszen a VIDEOGENIE 1 az 1977-ben (!!) piacra került TRS–80 modell minimális módosítású utánzata), devizaigénye magas (a Híradástechnika Szövetkezet elnökhelyettesének rádiónyilatkozata szerint az alkatrészek 2/3 részét kitevő import értéke 250 US $), és nem felel meg az iskolaszámítógép-pályázat feltételeinek (az eddig felsoroltakon felül még nem is „modulárisan bővíthető", és a szabad operatív kapacitás kisebb mint 16 kbyte).

Mivel az általam leírtak a laikus olvasók előtt nem ismertek, és így a 7. kín alapján a gépről jobb véleményt alkotnak, mint ami a valóságnak megfelelne, ezért kérem ezen levél helyreigazító célú közlését.

Dr. Simonyi Endre
1125 Budapest, Trencsényi u. 19.

Valutakidobás?

A Híradástechnika Szövetkezettel nem volt szerencsém. Hiába telefonáltam naphosszat, egyik vezetővel sem sikerült találkozót megbeszélnem. Pedig szerettem volna megkérdezni: miként nyerte el a Művelődési Minisztérium pályázatát a Híradástechnika Szövetkezet HT 1080Z iskolaszámítógépe?

A szövetkezet több százat gyártott és adott el eddig, s ez a gép szinte az összes középiskolában megtalálható. Rajta tanul a felnövekvő nemzedék. Ám a gépről megoszlanak a vélemények.

...A legmellbevágóbb közlés: e távol-keleti gyártmány túlságosan drága, teljesen elavult típus (hiszen a VIDEOGENIE 1 az 1977-ben piacra került TRS 80 modell minimális módosítású utánzata), devizaigénye nagy.

A Híradástechnika Szövetkezet elnökhelyettesének rádiónyilatkozata szerint a kétharmad részt kitevő import értéke 250 dollár. S ha ez így van, akkor miként felelhetett meg az iskolaszámítógép feltételeinek? Tény, az ára 35 ezer forint körül van, tévékészülék nélkül. Ám egészen másként cseng, ha hozzáteszszük, hogy minden egyes HT komputer 250 dollár értékű beépített alkatrészt tartalmaz. Pedig a pályázat kiírása szerint „korlátozva van a dollárért kapható alkatrészek beépítése a készülék összértékének 8 százalékára". Akkor itt

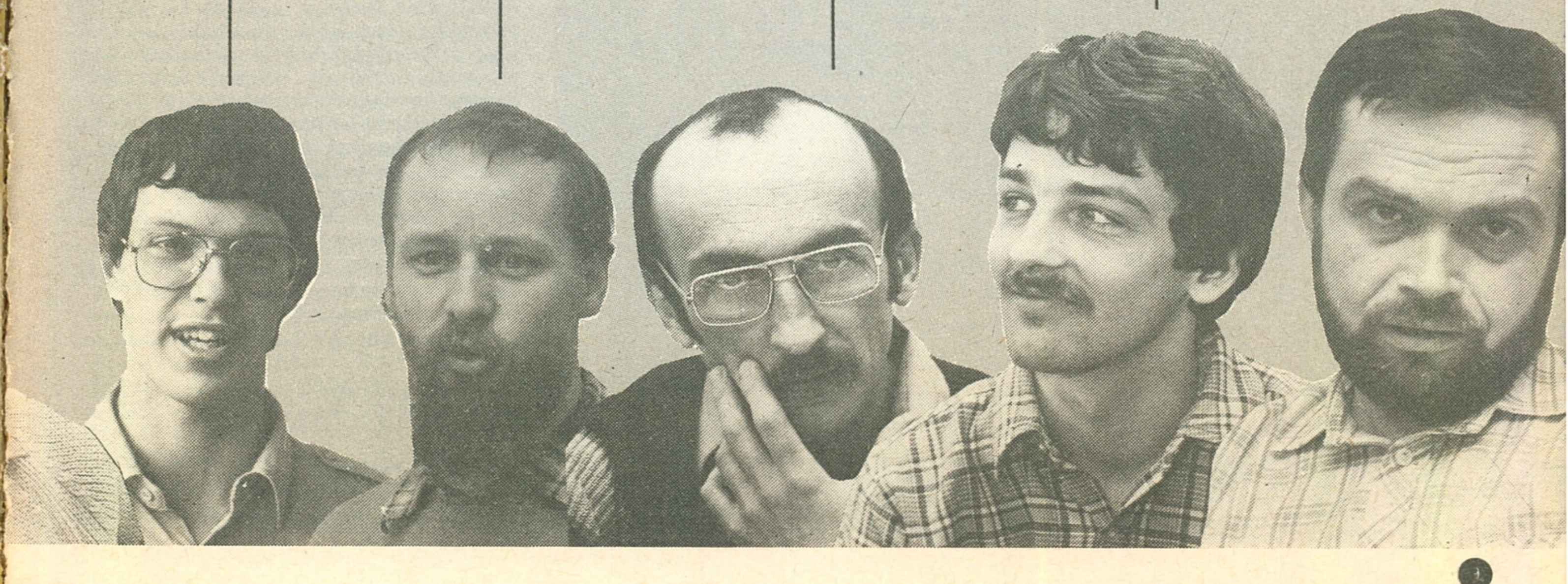
CZAKÓ TAMÁS: Még egy programom sem „szállt el"; nem tudom, ez mennyire általános tapasztalat!

TÖRÖK TURUL: Közel 2 éve használom, ezalatt sokkal „jobb" gépekkel összehasonlítva is hódolója maradtam!

TÓTH LÁSZLÓ

KIRÁLY ZOLTÁN

TÓTH FERENC: Ez a gép a szívem csücske. Sok klassz embert ismertem meg vele kapcsolatban. Úgy érzem, van még javítanivaló!



valami nagyon hibádzik. A másik tény: egy Commodore 64 típusú személyi számítógép ára 196 dollár (sőt, ma már olcsóbb is). Teljes képtelenségnek tűnik, hogy ily módon minden egyes HT 1080 Z legyártásával legalább 54 dollárt hajítanánk ki az ablakon.

A Híradástechnika Szövetkezet az olvasói levelekre nem válaszolt – s nekünk sem. Így amíg nem jön össze a randevú, s nem győznek meg az ellenkezőjéről, kénytelen vagyok elhinni, hogy szórjuk a dollárt a szupergépnek jóindulattal sem nevezhető masinára.

Cseszák Gyöngyi

(Magyar Hírlap/1984 március)

Válaszlevél

Tisztelt Szerkesztőség!

A lapjukban megjelent rovatolt cikkre eddig nem válaszoltunk, mert úgy véltük, hogy egy olyan hozzászólás felett, amely elfogult, nem tárgyilagos, indulatok hevében született és nem a problémák, gondok megoldását célzó szándékkal íródott, nem érdemes a széles körű nyilvánosság bevonásával vitát folytatni.

Nem tartottuk szükségesnek a választ azért sem, mert Önök tárgyilagosan közölték, hogy dr. Simonyi Endre „Simon '68" nevű mikrogépével érintett volt a pályázaton, s ebből a hozzászólás hangvételének egyik okára következtetni lehet. Úgy véltük, hogy akik a gépet használják, azoknak a választ a géppel kapcsolatos tapasztalat jobban megadja, mint egy esetleges magyarázkodásnak tűnő írásos válaszunk. Tekintve azonban, hogy a BIT-LET-ben megjelent és tárgyi tévedéseket tartalmazó „Hozzászólás"-ra támaszkodva a Magyar Hírlap „Valutakidobás?" címmel egy újabb cikket közöl, és kontroll nélkül olyan elmarasztaló következtetésekre jut, amelyek bántóak és rontják hitelünket – válaszunk elkerülhetetlen.

Válaszunk dr. Simonyi Endre levelében felvetettekhez igazodó sorrendben történik.

A gép megbízhatóságát illetően a felhasználói és a szerviztapasztalatok egyértelműen azt támasztják alá, hogy a gép eleget tesz az elvárásoknak, és nem rosszabb, mint a hazánkban előforduló tőkés személyi számítógép állomány bármely tagja. Ezt igazolják azok a vélemények is, amelyeket az oktató pedagógusoktól kaptunk.

A kazettás tároló első kísérleti gyártásánál voltak problémák. Ezeket a segítő szándékú konkrét észrevételek nyomán megszüntettük. A teljesség kedvéért jegyzem meg, hogy a kazettás tároló nem része egy számítógépnek. Annak perifériája. Beépítése nem volt követelmény. Beépítettük azért, hogy az iskolákban ne kelljen a csatlakozó kábelekkel bajlódni, és ne okozzon gondot a magnetofon külön történő beszerzése. Akkor nem gondoltunk arra, hogy ezzel a lépésünkkel egyeseknek támadási felületet teremtünk, és a pluszként beépített – nem általunk gyártott – periféria hibáját a számítógép hibájának minősítik.

A géphez a felhasználó igénye szerinti további külső magnetofon is csatlakoztatható.

Megbízhatóságot illető általános filozofálásra nincs válaszunk, azok felvetésének módja nem vitaképes!

A magyar ABC a pályázatnak nem volt követelménye! Ilyen igény az első széria alkalmazásba vételét követően merült fel, amelyet a második szériánál teljesítettünk is. A cikk első megjelenésekor már 1000 db gépet magyar ABC-vel szállítottunk. Ez válasz arra, hogy „hogy kaphatott egy olyan, iskolai célra államilag terjesztett gép jó minősítést, ami pl. nem ismeri a magyar ABC-t".

A gép dokumentációját (helyesen 3 kötetes felhasználói könyvet) elég szűkszavúan és nem valami hízelgően értékeli. Nem vitatom, hogy a három kötetből álló, közel 150 oldalas anyagban előfordulnak hibák. De jobban örültünk volna annak, ha nem egy szóval minősíti, hanem konkrétan rámutat magukra a hibákra. E gondolatkörnél is hivatkozik a pályázatra, amely azt támasztja alá, hogy azt nem gondosan olvasta el.

Lehet, hogy dr. Simonyi Endre egy iskolaszámítógéppel szemben magasabb igényeket támaszt, mint maga a pályázati kiírás. És arról az iskolaszámítógépről, amelyet egy széles körű zsűri a pályázati követelményeknek megfelelőnek tartott, ő kijelenti: „... – ez a gép iskolaszámítógépnek, így ahogy van, alkalmatlan", „...és nem felel meg az iskolaszámítógép-pályázat feltételeinek".

Erre csak azt tudjuk válaszolni, módunkban volt néhány olyan versenyen és a Neumann János Számítástechnikai Egyesület által szervezett programcserén részt venni, ahol 20–30 gép mellett ülő fiatalok lelkesen dolgoztak a gépeken. Azokat szeretik és olyan célokra is alkalmazni tudják, amire nem is gondoltunk. Tanáraikkal együtt elismeréssel adóznak a számítógépekről.

A legsúlyosabb tárgyi és tartalmi tévedés az árat, alkatrészt és a típust illető fejtegetése hordozza.

Ezzel összefüggésben elöljáróban leszögezem:

a) Az anyagárakat a késztermékárakhoz csak azonos időszakban lehet hasonlítani!

1982. évi anyagárat nem lehet és nem szabad a 1984-es készülékárral összehasonlítani, főleg, ha ez utóbbi valótlan!



b) *Számítógép és számítógép között különbség van!*

c) *A felelőtlen és valótlanságokat is hordozó bírálatra mi nem „sértődtünk meg", mert elköteleztük magunkat a középiskolák számítógép-ellátására és arra, hogy biztosítjuk a szervizt, a javító-alkatrészeket tároljuk, a vevőket kiképezzük stb.*

Mi a valóság?

A 250 US $ alkatrész-beszerzési ár 1982-ben az első kísérleti szériánál volt igaz. Hogy ez magas vagy alacsony – nézőpont kérdése. Ebben az időben az ekvivalens TRS–80 modell III. /16 K RAM típusú tőkés gép ára a METRIMPEX Külkereskedelmi Vállalat által beszerzett ajánlat alapján (másolat csatolva) 1158 US $ volt. A hazai gyártással akkor azonos devizáért négyszer több gépet tudtunk adni az iskoláknak.

A pályázat meghirdetését követően egy éven belül a készüléket csak úgy lehetett kibocsátani, hogy az alkatrészek többségét tőkés piacról szereztük meg. Akiknek a termelésben a legcsekélyebb jártasságuk van, azok tudják, hogy a szocialista vagy hazai alkatrészek beszerzésének átfutása mennyivel hosszabb időt vesz igénybe. Ha csak hazai és szocialista alkatrészeket akartunk volna felhasználni, még ma sem lenne iskolaszámítógép!

Ma már a devizaköltség 100 $/db alatt van, és jövőre tovább csökken.

A kezdetben beépített tőkés alkatrészhányad eredményének tudható be, hogy ebben a kategóriában egy gyártáskultúra honosodott meg szövetkezetünkben, és 35 400 Ft-ért elő tudtunk állítani egy számítógépet.

Az előbbiek tükrében kell értékelni, hogy az indulásnál megérte-e a 250 US $ alkatrészhányad, és hogy ez magas volt-e vagy sem. Nem tudjuk, mire alapozza dr. Simonyi Endre: „...a hazaival azonos kiépítésbeni NSZK-beli kiskereskedelmi ára 500 DM körüli".

A CHIP folyóirat 1983. évi többi számában is egy VIDEOGENIE 1. készülék ára tv-vevő, illetve monitor nélkül 1000 DM körül van. Tv-vevővel, illetve monitorral, amellyel mi szállítjuk a gépet, a CHIP 1984. márciusi számának 135. oldalán 1248 DM. Lehet, hogy valamelyik kiskereskedő tönkrement és kiárusított néhány darabot 500 DM-ért, de ez nem összehasonlítási alap.

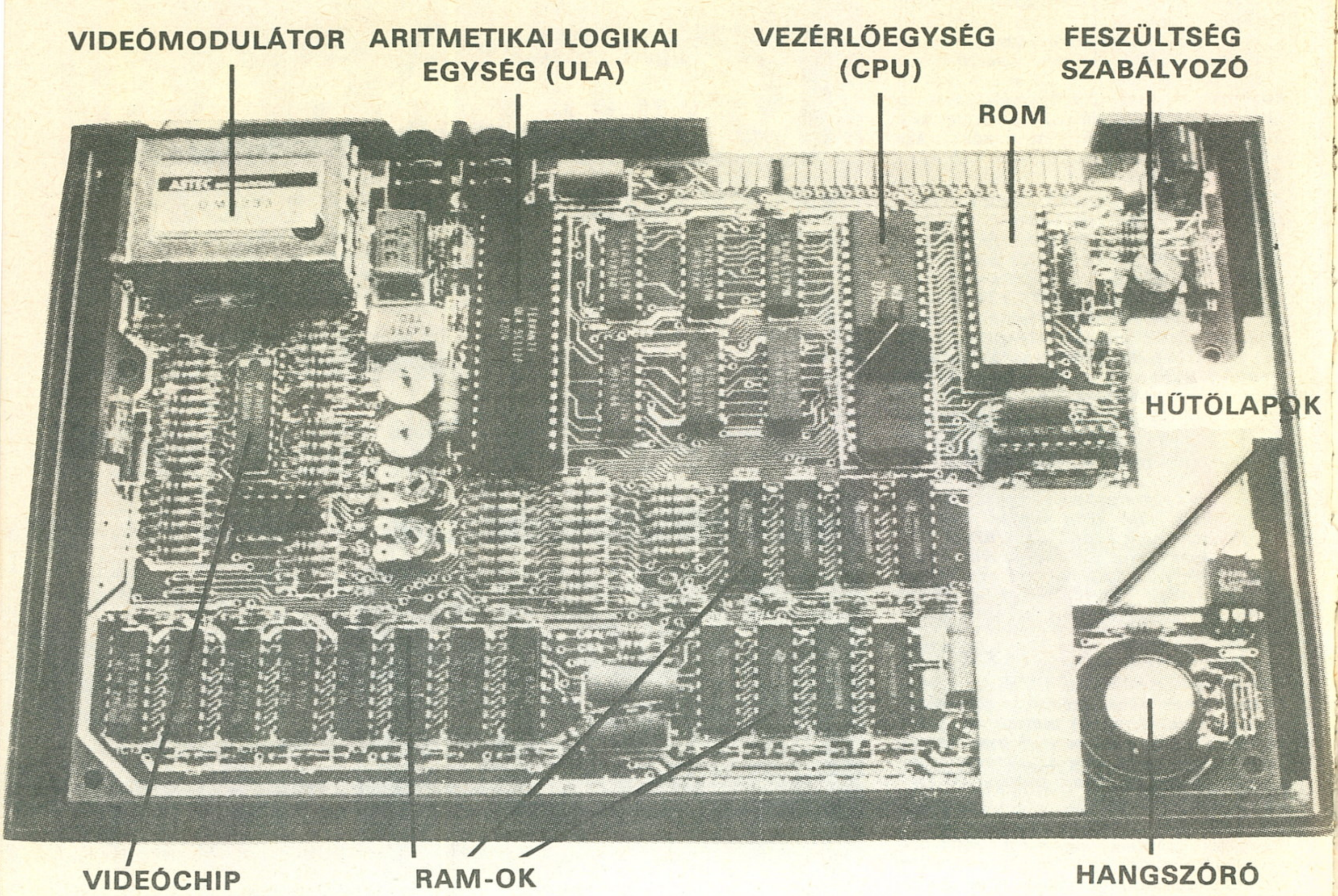
Hogy készülékünk egy teljesen elavult típussal ekvivalens, szintén nem helytálló, annál is inkább nem, mert akkor az 1984. évi nyugati folyóiratokban nem hirdetnék, és főleg nem ilyen áron, az iskolaszámítógépünkhöz hasonló paraméterekkel rendelkező készülékeket.

Befejezésül dr. Simonyi Endrének arra a kérdésére válaszolunk, hogy mindezért, amit csinálunk, kijár-e a tisztelet. Szavai szerint: „Ezért? Tisztelet?"

Mi nem igényljük tiszteletet, különösen nem azoktól a hozzászólóktól, akiket nem a segítő szándék vezérel. De igenis igényeljük az építő jellegű észrevételeket, amelyeket azzal a szándékkal tesznek, hogy a következő széria még jobb legyen. Ennek tulajdonítható, hogy azonos áron a következő sorozatban gyártott készülék operatívtár-kapacitása 64 kbyte lesz.

Novák Ferenc **kereskedelmi vezető**

Szerkesztőségi lezárás: No comment



Volt egy kis félelem bennünk, hogy más gépek magas átlagosztályzatai után az igazán világszínvonalnak számító gépeknél mi lesz? Hajlandóak-e szőrösszívű *inkvizítoraink* magasabb osztályzatokat is adni. Félelmünk alaptalannak bizonyult, a ZX Spectrum – jó hírét igazolva – 4,5-ös, szinte hihetetlen átlageredményt ért el! Ezt látva, ismét meg kell fogalmaznunk a *Vallató* álláspontját: nem egymáshoz hasonlítjuk a gépeket, hanem azt próbáljuk osztályozni, hogy a saját kategóriájában, a saját áráért mit nyújt egy-egy mikroszámítógép.

GYÁRI ADATOK

Csatlakozási lehetőségek: tv-készülék (színes vagy fekete-fehér), magnetofon, printer, I/O interface, beszéd szintetizátor, ZX microdrive.
Méret: 232x143x32 mm
Súly: 555 g

KÍNRENDSZER

Az eddig használt kínjaink itt is beváltak, egy újat azonban ismét érdemes volt felvenni a listára: amiről más gépeknél nemigen lehetett beszélni, azt érdemes a Spectrumnál osztályozni is, és ez a szoftverellátottság. A Vallató kiértékelés közben éreztük mégis, hogy a Spectrum sok helyen kilóg a kínrendszer szempontjaiból, nehéz osztályozni! Nem véletlen!
Ahogyan a Sinclair konstrukciójú más gépek is kilógnak a világ összes jól bevált, kiforrott konstrukciójú gépei közül – úgy feszegeti forradalmi, zseniális újításaival a mi kínrendszerünket is a Spectrum. De ez már kiderül a Vallatásból.

Ár – 4,8

1 KÍN

Ismét tettünk egy kísérletet arra, hogy külön osztályozzuk a külföldi és hazai árat, ismét nem sikerült. A külföldivel nincs különösebb baj, a kapott átlagosztályzat mutatja, hogy lenyűgözően olcsó a nyugati piacon. Az itthoni osztályzat problémásabb, elsősorban inkvizítoraink vérmérsékletétől függ. Azt mindenki elismerte, hogy itthon is ez a legolcsóbb, már majdhogynem forintban is megéri az árát, azonban – ismétlem vérmérséklettől függően – ki vérlázítónak, ki rablásnak, ki pedig egyszerűen pimaszságnak tartja, hogy mégis milyen sokba kerül itthon. Így egyestől ötösig a szivárvány minden színében tündököl a Spectrum (ára). Csak egy a közös: mindenki vörös a méregtől, ha erre gondol!

Perifériák – 3,9

2 KÍN

Itt máris érdemes szót ejteni a Spectrum konstrukciójáról. A megszokott megoldások helyett az alapgép és a perifériák is olyan műszaki ötleteket tartalmaznak, amelyekről már az egyetemen tanítják a jövendő mérnököknek, hogy: „Csak így ne, mert lehetetlen!" A Sinclair cég gondosan ügyel, hogy a mások által csak drágán megoldható dolgokat olcsón állítsa elő. És az észbontó az, hogy ezek a berendezések a fizika és az egyetemi professzorok szabályait felrúgva működnek! Hát valahogy így vagyunk a perifériákkal is. Az alapgéphez közvetlen csatlakoztatási lehetőség alig van. Mégis a Spectrum az, amelyhez a legtöbbféle periféria kapható. Egyik inkvizítorunk így fogalmazott: „Ma már csak a szoba mérete határozza meg, hogy hány kiegészítés köthető a Spectrumhoz!" Van saját nyomtatója – igaz, fémpapírral dolgozik, de mindössze 40 fontba kerül. Van „microdrive" (végtelenített szalagos háttérmemória, floppy gyorsasággal), még olyan „őrültségek" is elérhetők, mint a beszédszintetizátor vagy digitalizáló. Még egy, a perifériákhoz kapcsolódó megfogalmazás, ami a Sinclair cég piaci taktikájára talán a legjellemzőbb: „Nevetséges ez a nyomtató! Két kötőtű szaladgál benne összevissza! De nyomtat!"

Képernyőkezelés – 4,5

3 KÍN

Több előny és több hátrány. Az előnyök: ezért a pénzért nagyon szép grafikát képes csinálni. Külön előny a programból kezelhető FLASH és a fényesség. Hátrány: a hat szín közül három nagyon hasonló, s egy karakterpozícióban mindössze két szín használható.

ZX SPECTRUM

Hang – 3,6

4. KÍN

A ZX-eknél szokatlan módon, semmi zseniális nincs benne. Legfeljebb ennyi: zseniálisan egyszerű. Egy szólamban, könnyen programozható, kicsit halk is, kár, hogy a hangszóró alul van. Ennek ellenére létezik Spectrumra beszélő sakkprogram, ami azt mutatja, hogy elszánt programozóknak nincs akadály – írta az egyik inkvizítor. Hátrány, hogy a hang generálása leköti a processzort. Így közben semmi mást nem tud csinálni.

Kazettás tárolás – 4,7

5. KÍN

Valaki így fogalmazott: „Eddig kétszer nem sikerült programot bevinnem. Mind a két alkalommal az összekötő zsinór hiányzott!" Van persze más vélemény is, hosszabb programokkal már többen bajba kerültek, de általában mindenki megbízhatónak, jónak tartja.

Gépi kódú programozás – 4,4

6. KÍN

Valaki ezt írta: „Picit hiányzik a monitor". Egy másik vélemény: „Ezért az árért monitort követelni – illetlenség". A gépi kódú programozás lehetséges, de kissé nehézkes – ez volt a véleménye mindenkinek. Azonban rögtön hozzátették azt is, amiről eddig más gépeknél nem beszéltünk, hogy viszont olcsón kaphatók kitűnő fordító programok, így tulajdonképpen könnyedén programozható gépi nyelven is. Magától értetődő, hogy a gépi kódú program is „save"-elhető.

Megbízhatóság – 4,4

7. KÍN

Van, néhány a gyárból már hibásan kikerült gép. Ami azonban egyszer működött, annak még nem volt ideje elromlani. Inkvizítoraink közül eddig egy látott használatban elromlott gépet. A megbízhatóságra is a Sinclair-stratégia jellemző: „Vannak a gépben csúnya forrasztások? Vannak! De működik! Melegszik? Melegszik. De működik!"

Billentyűzet – 3,8

8. KÍN

A ZX 81 legnagyobb hibáját huszárvágással oldották meg: az olcsó fóliaérintkező megmaradt, csak van rajta egy gumi nyomógomb, ami majdnem olyan, mintha mozogna. Még a komoly billentyűzethez szokott kezek tulajdonosai is azt vallják, hogy „nem is olyan rossz!" Feltétlen előnynek tűnik, hogy egy gomb egy teljes utasítást visz be a programba, emiatt viszont egy-egy nyomógomb 6 funkciót is kezel. Így kezdetben a használata igen bonyolult, később azonban igen gyors. Valaki azt írta: „Aki vakon tud gépelni, az megbolondul!" A Sinclair cég ezt is válaszolhatná: „Gépelni lehet vakon, de programozni: szemtelenség!"

Dokumentáció – 4,6

9. KÍN

Ebben a kínban, ahogyan eddig is tettük, csak a géphez tartozó dokumentációt értékeljük. Ez a Spectrumnál azért is nehezebb, mert folyóiratok, klubok működnek szerte a világon, tehát a gépkönyv mellett igen sok egyéb publikáció is létezik. Önmagában a gépkönyv is lenyűgöző. Okos ötlettel az Oxfordi Egyetemet kérték fel a gépkönyv megírására, így mind pedagógiai, mind szakszerűségi szempontból kitűnő. Valaki ezt írta: „Tanítani kellene!" A gépkönyv mellé olyan demonstrációs kazettát is adnak, amin a billentyűkezelés tanulható, ami igazán kedves figyelmesség.

Editálás – 4,4

10. KÍN

Könnyen elsajátítható és kényelmes. Hibája, hogy a cursor nem mozgatható szabadon és hogy nincsen újrasorszámozás. Előnye viszont az azonnali szintaktikai ellenőrzés.

A gép programnyelve – 4,6

11. KÍN

Változatos, könnyen megérthető, az átlagnál jobb – ez az általános vélemény. Persze hibái is vannak: hiányzik az ELSE utasítás és a RENUMBER, viszont nagyon jó a MERGE és a SCREEN$.

Tanulhatóság – 4,6

12. KÍN

Ebben a kínban a már említett hibák és előnyök összegződnek. Általában jól tanulható gépnek tartják, hasznos a gépkönyv és a hozzá adott demonstrációs kazetta. A más gépekhez szokottak nehezen tanulják meg a billentyűzet sok funkcióját. Programozásnál azonban kifejezett előny, hogy a teljes utasításkészlet állandóan ott látható a billentyűkön, így a ritkábban használt megoldások is könnyebben eszébe jutnak a programozónak.

Emberközelség – 4,9

13. KÍN

Az igen magas átlagosztályzat alátámasztja az általános véleményt: nagyon barátságos jószág. Ez ismét a Sinclair-stratégia győzelme: egy elegáns, hűvös angol úr tudása egy hobó bohémságával keverve.

Szubjektív vélemény – 4,9

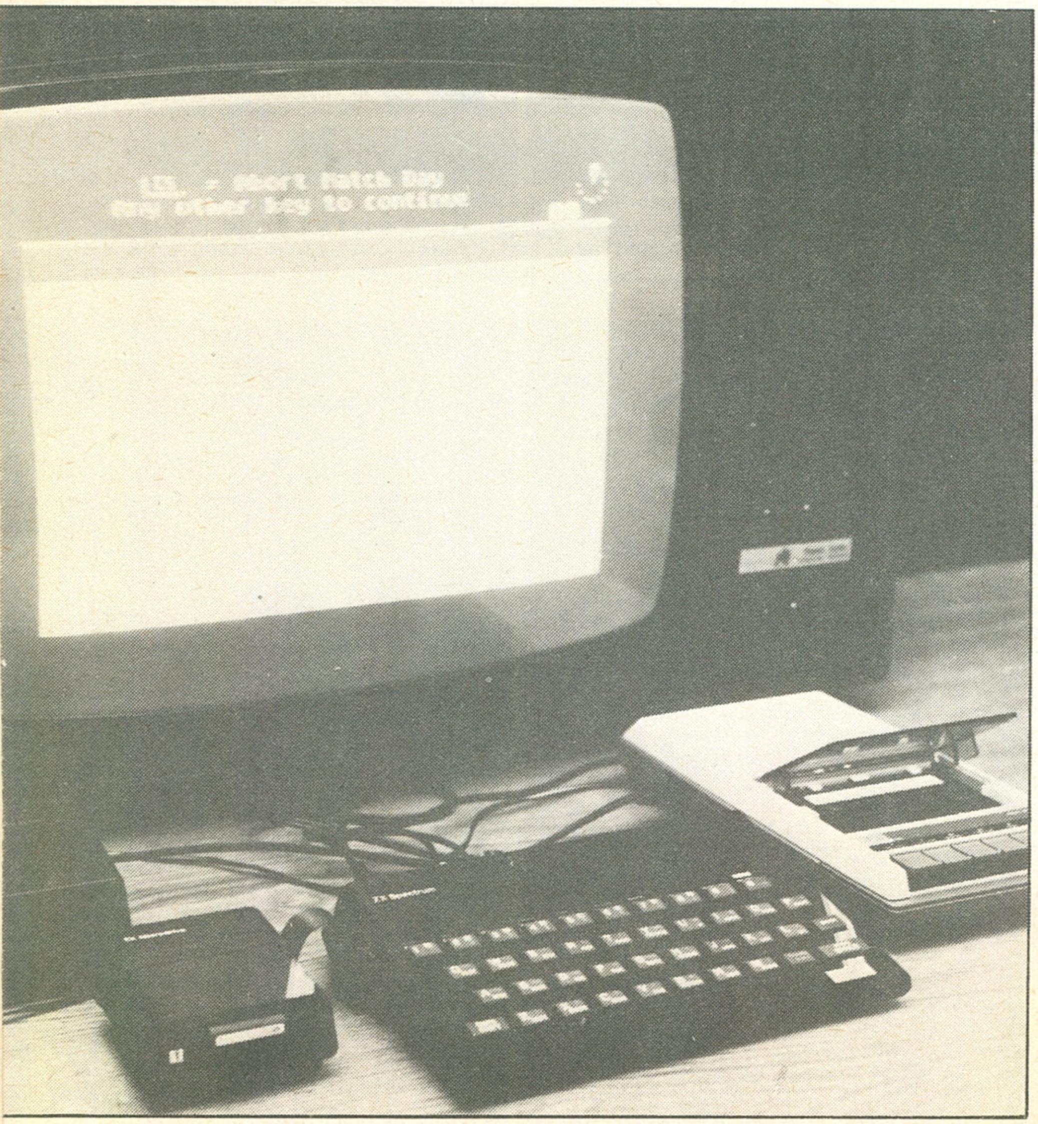
+1 KÍN

Az eddigiek alapján egyáltalán nem meglepő a magas átlagosztályzat. Különösebb jóstehetség nem kell ahhoz, amit az inkvizítorainkkal folytatott beszélgetésekből leszűrtünk: ennél magasabb osztályzatot egyhamar nemigen fog más gép kapni. Jobb gépek persze vannak, de mégis ez a kedvenc!

A ZX SPECTRUM VALLATASANAK EREDMENYE

KINOK	DR. SZEKELY JENO FOISKOLAI DOCENS	SANDOR ANTAL EGYETEMI OKTATO	SZINETAR JANOS FEJLESZTOMERNOK	KISS DONAT VILLAMOSMERNOK	KOVACS MIHALY KOZEPISK. TANAR	CZAKO TAMAS KOZEPISK. DIAK	DR. TOROK TURUL MATEMATIKUS	HUBERT TIBOR KOZEPISK. TANAR	DR. DROZDY GYOZO TUD. MUNKATARS	ATLAG
1 .KIN:AR	4	5	5	5	5	5	5	4	5	4.8
2 .KIN:PERIFERIAK	4	3	5	3	4	-	4	3	5	3.9
3 .KIN:KEPERNYOKEZELES	4/5	5	4/5	4	5	4	4/5	5	4	4.5
4 .KIN:HANG	3	4	3/4	3	3	4/5	4	3	4	3.6
5 .KIN:KAZETTAS TAROLAS	4/5	5	5	4/5	4	5	4	5	5	4.7
6 .KIN:GEPI KODU PROGRAMOZAS	4	4	5	-	4	5	4/5	4	5	4.4
7 .KIN:MEGBIZHATOSAG	5	4	5	5	4	5	4	4	4	4.4
8 .KIN:BILLENTYUZET	4	3/4	4/5	4	3	4	4	3	4	3.8
9 .KIN:DOKUMENTACIO	5	3/4	5	5	5	5	4	4	5	4.6
10 .KIN:EDITALAS	4	4/5	4/5	5	4	5	4/5	4	4	4.4
11 .KIN:A GEP PROGRAMNYELVE	4/5	4	5	4/5	5	4/5	5	4	5	4.6
12 .KIN:TANULHATOSAG	4/5	4	5	5	4	5	4/5	4	5	4.6
13 .KIN:EMBERKOZELSEG	4/5	5	5	5	5	5	5	5	5	4.9
+ 1 KIN:SZUBJEKTIV VELEMENY	4/5	5	5	5	5	5	5	-	5	4.9
+ 2 KIN:SZOFTVER ELLATAS	5	-	5	5	5	5	4	-	5	4.9
ATLAG	4.3	4.3	4.8	4.5	4.3	4.8	4.4	4.0	4.7	4.5





Szoftverellátottság – 4,9

+2 KÍN

A Spectrum már elmondott tulajdonságaiból és igen alacsony árából következik, hogy az elmúlt évek egyik legkedveltebb gépe lett. Becslések szerint az angol felhasználók 20%-a ezzel a géppel dolgozik, ami azt jelenti, hogy a legelterjedtebb típus. Ebből következik, hogy a Sinclair cégen kívül mások is foglalkoznak perifériák gyártásával és az is, hogy a programkazetták, szoftvercsomagok legjobban ehhez a típushoz vásárolhatók. Ez az elterjedtség nagyon megnöveli a felhasználás sokoldalúságát, mert talán éppen a Spectrum lehet majd az a gép – amennyiben még nem az – amit a háziasszonyok és a farmerek is megvesznek. Olyan emberek tehát, akik maguk nem tudnak és nem akarnak programozni, de használni tudnák a gépet.

Kissé egyoldalúnak tűnhet az olvasónak a VALLATÓ, hiszen alig van benne más, csak áradozás a Spectrumról és a Sinclair cégről. Ennek alapján talán mindenki szaladhatna a boltba, hogy beszerezze a szükséges Spectrumokat. Visszatartani senkit nem akarunk, de a reális értékeléshez érdemes megismételni a Vallató alapszabályát: a saját kategóriájában, a saját áráért ilyen jó a Spectrum. És még valamit! Inkvizítoraink magas osztályzataikkal lerótták tiszteletüket a Sinclair cég újításai, ötletes megoldásai előtt is. Erre a legjellemzőbb Török Turul néhány mondatban megfogalmazott szubjektív véleménye (ami persze a Spectrum további dicsérete is) és amit mi is vállalunk:

Csak az állandók változnak
a változók soha!
(Osborn törvénye)

ZX SPECTRUM



„A számítástechnika népszerűsítését egyedülállóan szolgáló »cégecske« szinte diktálja a feltételeket a nagyoknak. Olcsóságát nem az eladhatatlan készletek (mint a TI 99/4A vagy VIC 20 esetében), hanem a zseniális műszaki megoldások eredményezik. A ZX 80–81 gyerekbetegségeit nagyrészt orvosolták és a kezdő réteget megcélozva olyan játékszert bocsátottak ki, amit felnőtté válva sem fog eldobni tulajdonosa, inkább megszállottként védelmezi majd."

HOZZÁSZÓLÁS

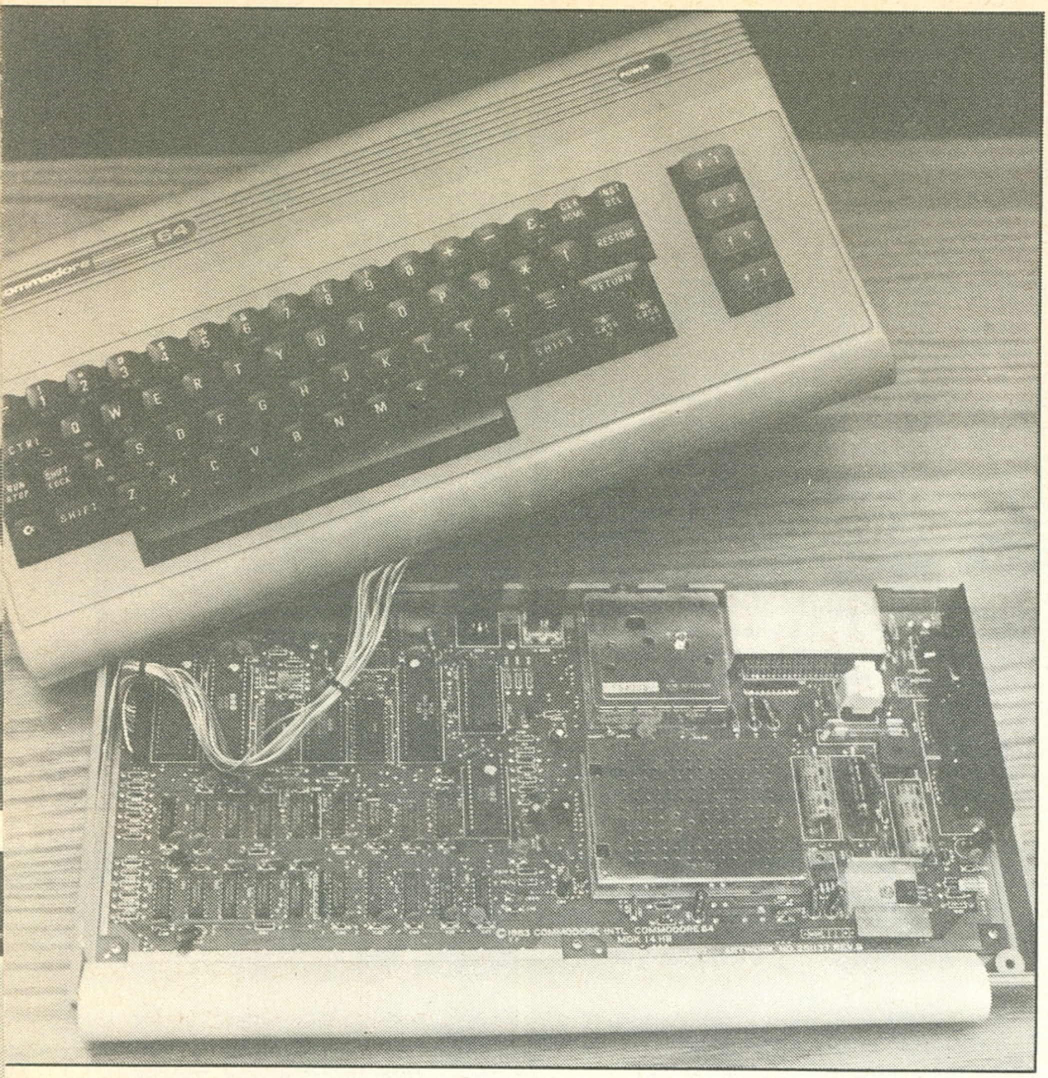
Tisztelt Szerkesztőség!

A BIT-LET-ben a Commodore 64 személyi számítógép szerkesztőségi tesztelése közben olyan megjegyzéseket tettek a Spectrum gépekre vonatkozóan, amelyek – ha bizonyos ellentmondásoktól el is tekintünk – nem fogadhatók el megalapozottnak. Bizonyára a megfontolt értékeléseken túl sokakat érzelmi kötődések is motiválnak. Így könnyű lehet heves vitákat kiprovokálni a különböző géptípusok hívei között. Nem célunk erre alkalmat adni, hanem néhány olyan pontra mutatni, amelynek ismeretében még tarthatatlanabb a Spectrum ilyen nagyvonalú leírása az egyszerű játékokon túlmutató alkalmazások területéről.

Az egyik ilyen pont az, hogy a személyi számítógép kategóriában a személyi használathoz kapcsolódó követelményeket kikezdi az a valóságos mikrogép-hiány, amely annyira jellemző, sőt nyomasztó. A vállalat, a hivatal érthetően mohó – minél több kapacitás, minél több floppy, minél robosztusabb gép kell neki. Észre sem veszik, és már semmi perszonális sincs abban a kemény szolgálatba fogott „konfiguráció"-ban. Nos, látni kell, hogy jogosak a közületi igények is, de jogos az is, hogy minél szélesebb értelmiségi vagy éppenséggel vezetői réteg használja személyesen a személyi gépet, akár a sajátját, akár a szolgálatit. Ilyen használatra, figyelembe véve a realitásokat (árakat, beszerzési lehetőségeket, stb.) ma a Spectrum az egyik legjobb személyi számítógép.

A másik dolog, amire felhívnánk a figyelmet, hogy – viszonyítások nélkül, a gép sajátos lehetőségeit vizsgálva – a Spectrum gépek rendkívül igényes módszerek alkalmazását teszik lehetővé, amire példák lehetnek a Kereskedelmi Szervezési Intézetnél kidolgozott közgazdasági rendszerek is. Ezek már most, jelenlegi formájukban is sikerrel alkalmazhatók nemcsak keserves manuális számítások gépesítésére, de a vállalati közgazdasági elemző-tervező munka minőségi javítására. Ezzel pedig meggyőződésünk szerint még távolról sem merítettük ki a Spectrum lehetőségeit.

dr. Simeonov Tódor, dr. Bódis Béla



Van egy gép, amelyről előző találkozásainkkor inkvizítoraink sokatmondó pillantásokkal, talányosan csak ennyit mondtak: „Majd a Commodore!" Dicsérhette bárki a Spectrumot, a HT-t, akármit, valaki mindig akadt, aki az említett pillantással csak a fejét csóválta. Majd a Commodore! Majd akkor meglátjuk, mi is az a számítógép! Így azután komoly várakozással kezdtünk az 1983–84-es év gépének kikiáltott Commodore vallatásához. És már az elején hozzátehetjük – bár az átlagosztályzat elmarad a Spectrumétól – nem csalódtunk!

GYÁRI ADATOK

Memóriaméret: 64 kbyte

Csatlakozási lehetőségek: tv-készülék (színes vagy fekete-fehér), magnetofon, lemezmeghajtó, printer, plotter, digitalizáló, ... szóval akármi.

Méret: 400x210x70 mm

Súly: 1,8 kg

KÍNRENDSZER

Az eddig alkalmazott szokásos kínjaink közül csak egyet változtattunk meg, az 5. számút: a kazettás tárolás megbízhatóságát. Ennél a gépnél ugyanis az elterjedt és egyértelműen jobb módszer a lemezes tárolás. Így az 5-ös kín megfogalmazása most a „tárolás megbízhatósága". A Spectrumnál már használt új kínt, a „szoftverellátást" most is alkalmaztuk, hiszen a Commodore-nál is – szerencsére lehet erről beszélni. Most nem voltak olyan gondjaink, mint ~~Spectrumnál~~, hogy úgy éreztük volna, hogy a gép kilóg az előre kitalált kínrendszerből. A Commodore szépen simul az elképzeléseinkhez, konszolidált, nyugodt konstrukciónak tűnik. Körülbelül az a különbség a Spectrum és a Commodore között ebből a szempontból, mint egy jóindulatú, de zavaros hobó és egy decens, nyakkendős üzletember között.

Ár – 4,8

1 KÍN

Az osztályzat jeles, de talán felesleges is azt leírni, hogy ez a külföldi árra vonatkozik. Talán meglepi azokat ez az eredmény, akik a Spectrummal összehasonlítják, így máris le kell szögeznünk egy alaptételt: bár mindkét gépet a „játék" kategóriában fejlesztették ki, mégis a Spectrum valóban csak játékra való, a Commodore ennél többet tud. Lényegében tehát inkvizítoraink úgy érezték, a két gép nem tartozik azonos kategóriába. Így talán érthetővé válik, hogy a magasabb ár is jeles osztályzatot kapott.

Természetesen külön téma a hazai ár, amelyre inkvizítoraink többsége hármast adott. Egy ötös osztályzat volt, de talán éppen ennek a magyarázata az elfogadható: „Magyarországi viszonyok között ez az ár ötös, de nevetséges, hogy ilyenek a viszonyaink". Nem bocsátkozunk ismét a kereskedelem és a vámszabályok szidásába, mert már kifogytunk a szitkokból, most inkább egy más, eddig nem tárgyalt szempontból közelítjük a hazai számítógép-árakat. Inkvizítoraink egyike megrendelőként végigjárt néhány hazai nagy- és kisszámítógép-gyártó vállalatot. Részben felmérésként, részben valóságosan egy a Commodore-hoz hasonló tudású gépet rendelt. A cégek első válasza az volt: „Kérem, keressen fel minket 1986 első negyedévében. Köszönjük". Inkvizítorunk is megköszönte és feltette a másik kérdést: mibe kerülne a gép? A válasz nem lepte meg: „200 000–300 000 forintért megcsináljuk". Inkvizítorunk ismét köszönetet rebegett és vásárolt egy Commodore-t, a hazai irreális áron. Ez a kis felmérés is igazolja: ilyen viszonyok között...

Perifériák – 4,9

2 KÍN

Többen vallották, hogy a Spectrumnál leírtak, tehát az, hogy „a perifériák számát csak a szoba mérete határolja be", a Commodore-ra talán még inkább érvényesek. A géphez szinte minden kapcsolható és – a Spectrumhoz csak ígéretként létező perifériákkal szemben – ezek bármikor megvásárolhatók is. Az egyetlen négyes osztályzat magyarázata, hogy a soros kapcsolású periféria mégsem az igazi, de ez is csak apró ellenvetés. Bosszantó viszont, hogy nem szabványos az interface kimenet. Inkvizítoraink erre egyetlen ésszerű magyarázatot találtak: mert így külön el lehet adni a csatlakozót! Erre a megállapításra a következő párbeszéd zajlott le vallatásunk kedves közjátékaként:

– Persze, sok cég csinál plusz egy bütyköt a csatlakozójára, csak azért, hogy külön el lehessen adni!

– Na és? A KGST-ben nem csinálják ugyanezt a csatlakozókkal?

– Miért? A KGST-ben van csatlakozó?

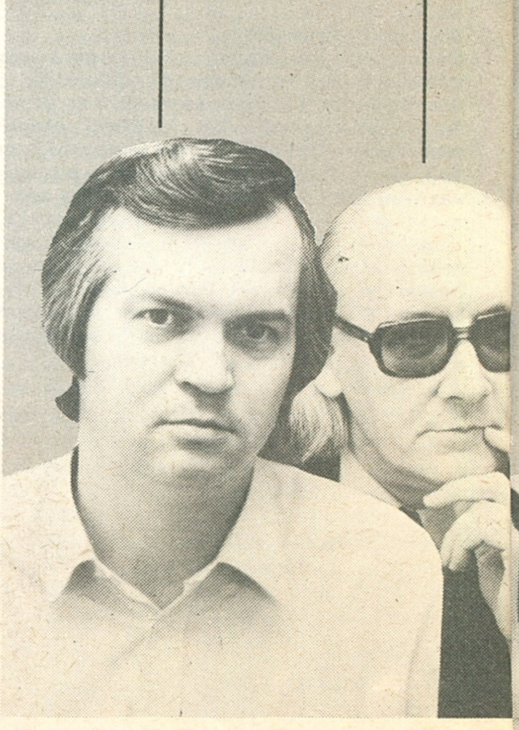
Képernyőkezelés – 4,8

3 KÍN

Inkvizítoraink véleménye nem tért el: a grafika kitűnő, lényegében mindent tud, igaz, jó szakember kell ahhoz, hogy kicsalja belőle. De ez már nem a gép hibája.

TÓTH FERENC: Hőmérsékleti körülményekre nem érzékenyebb, mint az ember!

LÁNGOS ISTVÁN: A gép éppen csak hogy meg nem szólal a kérdéseimre!



KÍNPADON A

COMMODORE 64

Hang – 4,9

4 KÍN

Három szólam, felfutás, lecsengés, rezonancia és hullámformabeállítás – és mindez hi-fi szinten. Egy ellenvetést hallottunk csak: nem tud annyit, mint a nagy MOOG szintetizátor. Hát igen.

Tárolás megbízhatósága – 3,8

5 KÍN

Megoszlottak a vélemények, ahogyan ez az osztályzatokból kiderül. Inkvizítoraink alapvetően a lemezes tárolást osztályozták, mert ez az elterjedtebb, fejlettebb – és nekik is leginkább erről volt információjuk. A szerviz képviselőjétől azt is megtudtuk, hogy gyári hibás lemezmeghajtó egységek is forgalomba kerültek, amit ugyan a gyár azonnal és díjmentesen kicserél, csak éppen nem tud róla, hogy Magyarországon is van 1500 példány. Ők ugyanis nem szállították ide. Aki viszont jó lemezmeghajtót fogott ki, annak azóta sem igen volt oka panaszra. Talán ez is indokolja, a vélemények szóródását.

Gépi kódú programozás – 4,1

6 KÍN

Az alapgépben nincs monitor funkció, ennek ellenére a többségnek az a véleménye, hogy fordító programmal kitűnően használható. Egyik inkvizítorunk még ezt is megkockáztatta: „Azért nem csinálták tökéletesen, nehogy kiszorítsa a nagyobb, profiknak szánt Commodore gépeket!"

Megbízhatóság – 4,6

7 KÍN

Az osztályzat miatt nem kell szégyenkeznie. Egy inkvizítorunk nem adott érdemjegyet, mert úgy érezte, hogy fél év alatt még nem volt ideje a gép megbízhatóságát megismerni. A hármas magyarázata az volt: „A júliusi 38 fokos meleget nem bírta. Izzadt". Erre csak egy válasz érkezett inkvizítorainktól: „– Te is!"

Billentyűzet – 5,0

8 KÍN

Az osztályzat egyértelmű, még egy csillagot is mellé tehettünk volna. Az írógéphez közelálló billentyűzet mindenkinek elnyerte a tetszését.

Dokumentáció – 2,4

9 KÍN

Hát ez bizony szégyen! Szerintünk! A tömör vélemény ez volt: hiányos, hibás, félrevezető. Gépkönyvként is használhatatlan, mert a lehetőségek nagy részét nem is említi, de BASIC tankönyvnek is csapnivaló. Inkvizítoraink egyike külföldön tárgyalt a Commodore képviselet vezetőjével. Megemlítette neki ezt a problémát. A válasz rövid volt: „Mi gépet gyártunk, nem könyvet!"

Editálás – 4,4

10 KÍN

Az osztályzat nem rossz, de az általános vélemény ez: komolyabb editor komolyabb gépet kíván. Többnyire mindenki elégedett a lehetőségekkel és többnyire mindenki hiányol néhány funkciót: kár, hogy nincs RENUMBER, DELETE és MERGE.

A gép programnyelve – 3,9

11 KÍN

Inkvizítoraink nem a legjobb véleménnyel voltak az alapgép nyelvéről. Sokféle megfogalmazást hallottunk: „erre nem lehet büszke!" vagy „nem lenyűgöző", de rögtön hozzátette azt is mindenki, hogy alapnak azért megfelelő. A számtalan különféle bővítési lehetőséggel pedig kitűnőnek is tartották néhányan. A gyártók talán tudják, inkvizítoraink közösen sem találtak magyarázatot néhány utasítás elhagyására (pl. ELSE).

Tanulhatóság – 4,6

12 KÍN

Az átlagosztályzat meglepően jó, ahhoz képest, hogy a géphez kapható dokumentáció milyen csapnivaló. Így tehát a gép könnyen tanulható, csak nincs miből. És ebből derül ki, hogy a gép felépítése, kezelése valóban olyan logikus és egyszerű, hogy még megfelelő használati utasítás nélkül is megérthető. Akkor pedig... tényleg az.

Emberközelség – 4,4

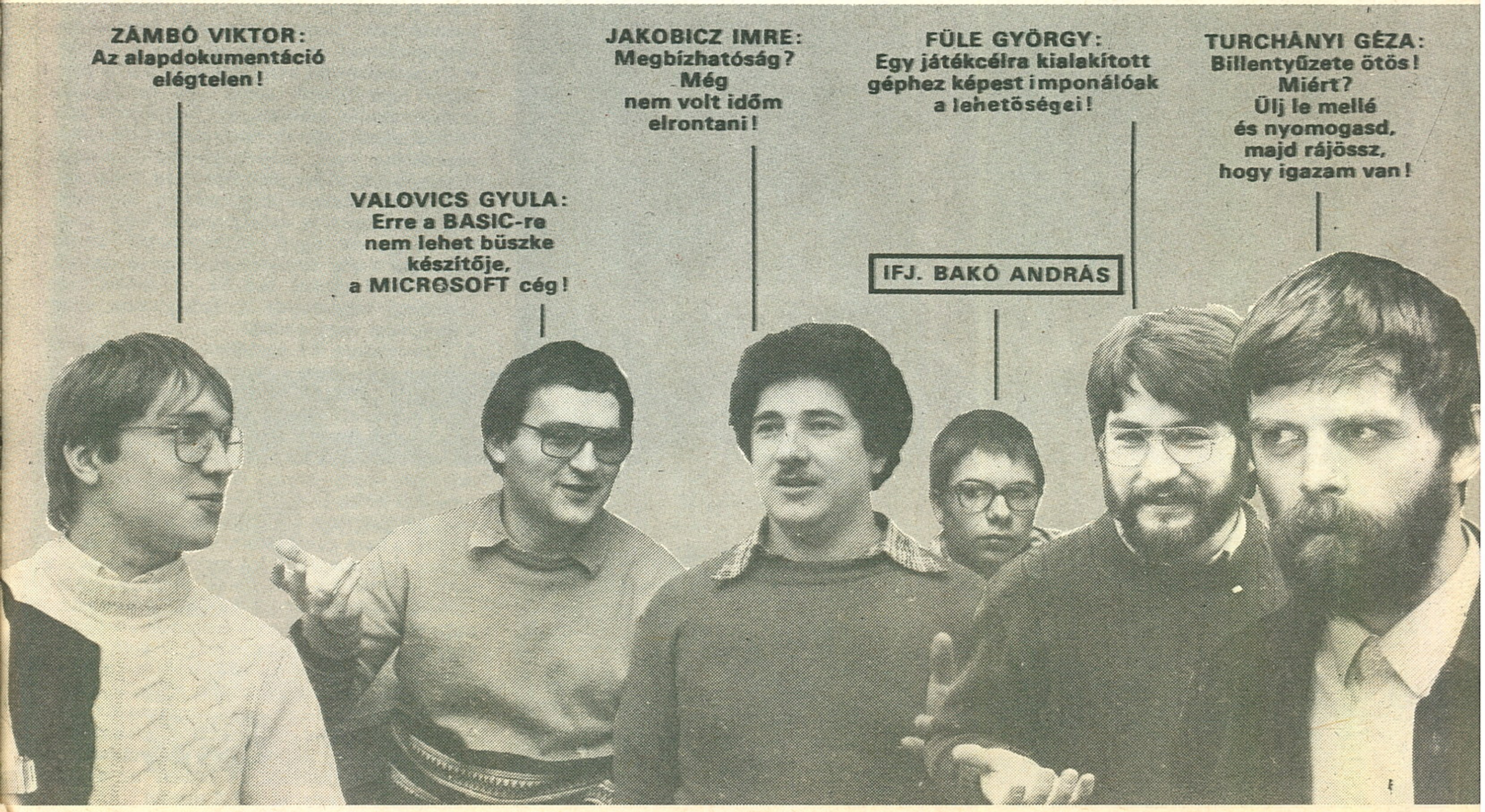
13 KÍN

Azoknak a felhasználóknak, akik BASIC programokat írnak, hang és grafika nélkül kitűnő. Ha viszont ennél többet is akar az ember, akkor bonyolódik a használat – ez volt az általános vélemény. Megfogalmazódott azonban ez is: „Túlságosan is emberközeli. Amióta otthon is van belőle – nem ismerem meg a feleségemet".

Szubjektív vélemény – 5,0

+1 KÍN

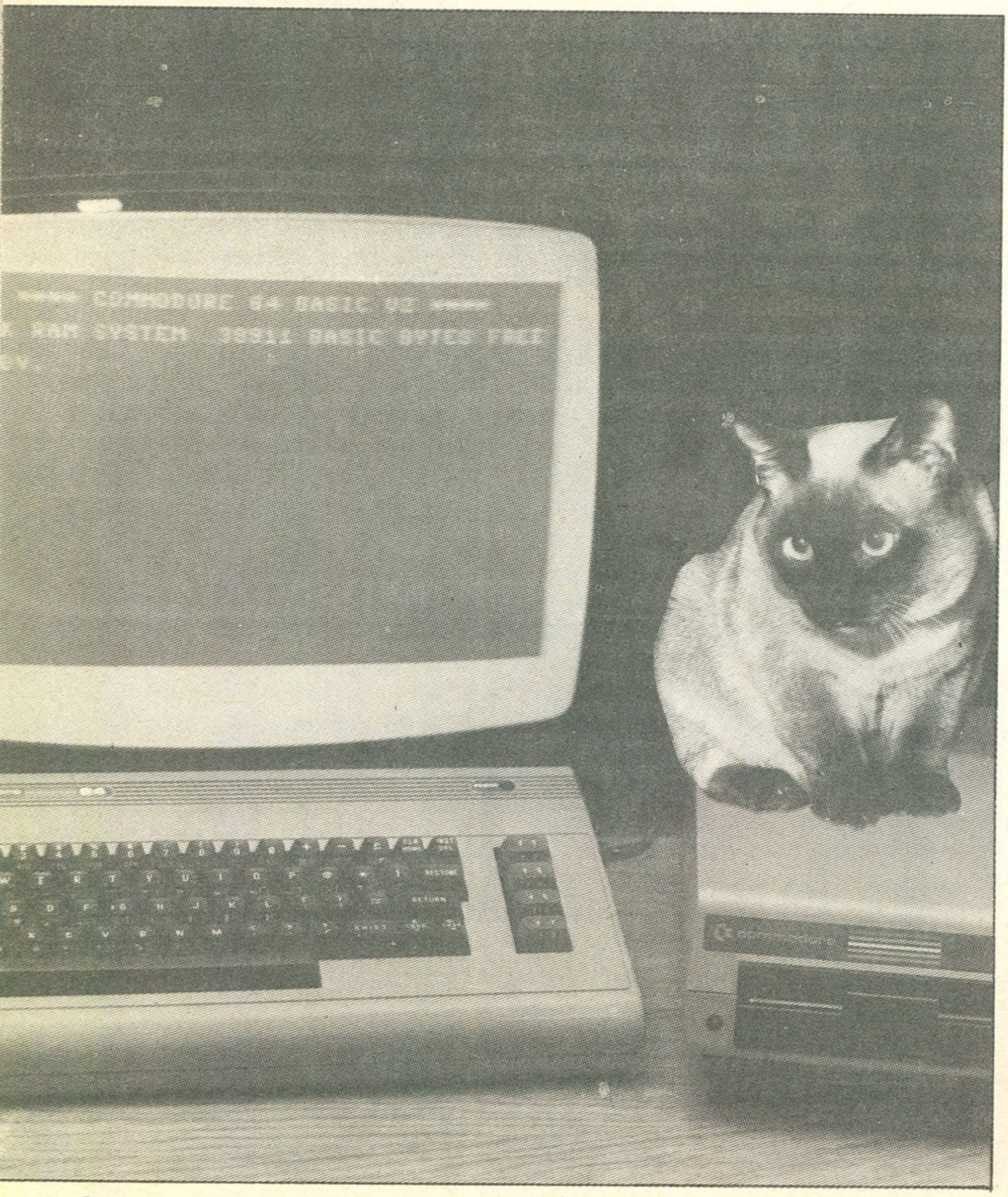
Emberközelség ide vagy oda, dokumentáció ide vagy oda – ilyen egyértelmű ötöst még nem kapott számítógép Válólató rovatunkban. És ez az osztályzat – úgy érezzük – többet mutat. Talán azt, hogy bár az egyes funkciókban lehet találni kivetnivalót, mindenki megszerette, kedveli a gépet, tudja használni és szívesen használja. És ennél többet kívánhat egy számítógép?



KÍNPADON A COMMODORE 64

A COMMODORE 64
VALLATASANAK EREDMENYE

KINOK	VALOVICS GYULA SZ.TECHNIKAI M.TARS	LANGOS ISTVAN SZ.TECHNIKAI FM.TARS	TOTH FERENC SZERVIZMERNOK	FULE GYORGY FIZIKUS	JAKOBICZ IMRE VILLAMOSMERNOK	ZAMBO VIKTOR EGYETEMI HALLGATO	IFJ. BAKO ANDRAS KOZEPISKOLAI TANULO	TURCHANYI GEZA KFKI MUNKATARS	ATLAG
1 .KIN:AR	5	5	5	5	5	4	4	5	4.8
2 .KIN:PERIFERIAK	4	5	5	5	5	5	5	5	4.9
3 .KIN:KEPERNYOKEZELES	5	4/5	5	5	[illegible]	5	5	5	4.8
4 .KIN:HANG	5	4/5	5	5	5	5	5	5	4.9
5 .KIN:TAROLAS MEGBIZHATOSAGA	4	3	5	5	4	2	4/5	3	3.8
6 .KIN:GEPI KODU PROGRAMOZAS	4	5	4	4	4	4	-	4	4.1
7 .KIN:MEGBIZHATOSAG	5	5	5	-	5	5	4	3	4.6
8 .KIN:BILLENTYUZET	5	5	5	5	5	5	5	5	5.0
9 .KIN:DOKUMENTACIO	-	2	3	3	2	2	2/3	2	2.4
10 .KIN:EDITALAS	4	5	5	4	4	4	4/5	5	4.4
11 .KIN:A GEP PROGRAMNYELVE	4	4	4	4	3	3	5	4	3.9
12 .KIN:TANULHATOSAG	5	5	5	4	3	5	5	5	4.6
13 .KIN:EMBERKOZELSEG	4	5	5	-	3	4	5	5	4.4
+ 1 KIN:SZUBJEKTIV VELEMENY	5	5	5	5	5	5	5	-	5.0
+ 2 KIN:SZOFTVER ELLATAS	5	5	5	4	4	4	-	4	4.4
ATLAG	4.6	4.5	4.7	4.5	4.1	4.1	4.6	4.3	4.4



Szoftverellátás – 4,4

+2 KÍN

Érdekes módon a hasonló kategóriájú „játékokra termett" gépek közül erre van a legkevesebb program. Az osztályzat mégis elég jó, aminek az a magyarázata, hogy a Commodore-ra viszont már sokkal bonyolultabb felhasználói szoftverek is léteznek. És itt már érintünk egy olyan területet, amiről az eddig tárgyalt számítógépeknél nemigen volt, mert nem lehetett szó. Azt, hogy milyen területeken, hol alkalmazzák a gépet, mire használható. A Spectrum vagy a HT a játékon kívül használható oktatóprogramok készítésére, vetélkedőkhöz és ezzel vége is a sornak. A Commodore 64 ezzel szemben szinte mindenre használható, amire egy komolyabb, professzionális számítógép. Rögtönzött körkérdésünkből kiderült, hogy használják mérés vezérlésére, nagy számítógéphez intelligens terminálként, szövegfeldolgozásra, folyamatvezérlésre, alkalmas adattárolásra, ügyvitelre stb. Inkvizítoraink valamennyien úgy érezték, hogy az egyik legfontosabb szempont: bár a gép arra készült, hogy játsszanak vele, sokkal többet tud. Ez az, amiért nem sorolható ugyanabba a kategóriába, mint a Spectrum vagy a többi.

A Commodore 64 esetében külön érdemes néhány szót ejteni a magyarországi felhasználási területekről. A mi viszonyaink között ez a mikroszámítógép átalakult, más szerepet játszik, mint amit eredetileg szántak neki. A gyártók játék-gépnek készítették, ám hazai magas ára egyelőre lehetetlenné teszi, hogy bárki ezért vásárolja meg. (Habár több külföldön tartósan dolgozó ezt hozta haza gyerekének.) A nagy gépekhez viszonyított alacsony ára azonban indokolttá tette, hogy más területeken a legjobban használható gép legyen. Mindez magyarázza azt is, hogy miért nem kell szégyenkeznie a végül is kapott 4,4-es átlagosztályzat miatt. Igaz, a Spectrum egy tizeddel jobbat ért el, de akkor egy játékgépet osztályoztunk, azzal az alapállással, hogy arra mennyire alkalmas. A Commodore-tól többet vár el a hazai felhasználó. Ezt a kisgépet már inkább a nagyobbakhoz hasonlítja a használója. És ebben az összehasonlításban érte el a 4,4 tizedet.



KÍNRENDSZER

Az ár, ami nincs

1 KÍN

A gép van is és nincs is. 1979 óta több száz darab található az országban, elsősorban iskolákban, egyetemeken; üzletben azonban nem vásárolható. A svédországi gyártást is leállították, a nemzetközi katalógusok már nem említik, így a jelenlegi árról beszélni nincs értelme. A gyártás idején a hazai ára 60–70 000 Ft körül volt, ami elég magasnak tűnik, de ne felejtsük el, hogy ez még a „nagy bum", a nagy számítógépár-csökkenés előtt volt. Így inkvizítoraink úgy döntöttek, hogy az árat nem osztályozzák.

Perifériák – 3,1

2 KÍN

Ebben a kínban az alapgéphez közvetlenül kapcsolható perifériákat szoktuk osztályozni. Inkvizítoraink többsége a kazettás rögzítőn kívül semmiféle más perifériával nem találkozott. Tudunk róla, hogy köthető a géphez minifloppy és sornyomtató, de az országban csak néhány helyen van ilyen berendezés. A második kínnál két pártra szakadtak inkvizítoraink: az egyik tábor, akik kedves emlékként gondolnak az ABC 80-ra, akik tehát azóta több újfajta gépen dolgoztak és lényegében ma már nem ülnek le mellé. A másik tábor az ABC-hívők, akik akkor megszerették, és ma is gyakran használják. Érthető, hogy a második tábor igyekezett a gép eredeti lehetőségeit jobban kihasználni, és az elmúlt néhány év fejlődését valahogyan megtanítani a kissé már koros mikroszámítógéppel. Így inkvizítoraink egy része meglepve hallotta, hogy egy furfangos, rövidke programmal már szinte bármilyen sornyomtató illeszthető a géphez, rajzolót is működtet és van olyan hely, ahol intelligens terminálként nagy számítógéphez használják. Az elért osztályzat mindezek alapján egy kicsit rosszabbnak tűnik a lehetőségeknél, viszont tartalmazza a való-

Az ABC 80 mikroszámítógépre nagyon illik a hasonlat: a népmesék hőse. Nevét legendák övezik, tisztes számítógépesek könnybe lábadó szemmel emlékeznek rá, van is meg nincs is, és talán majd egyszer visszatér, mint Rózsa Sándor. A mikroszámítógépek első típusai között volt, így a jelenleg dolgozó számítástechnikai szakemberek nagy része ezen tanulta a BASIC nyelvet, ezen a gépen ismerte fel, hogy a nagy és misztikus számítógépek egyeduralmának a korszaka lejárt. Talán ezzel magyarázhatók a hozzá kapcsolódó legendák és elérzékenyült programozók, akik az ifjúságukra emlékeznek.

GYÁRI ADATOK

Gyártó cég: Luxor, Svédország licence alapján Magyarországon a BRG gyártotta.

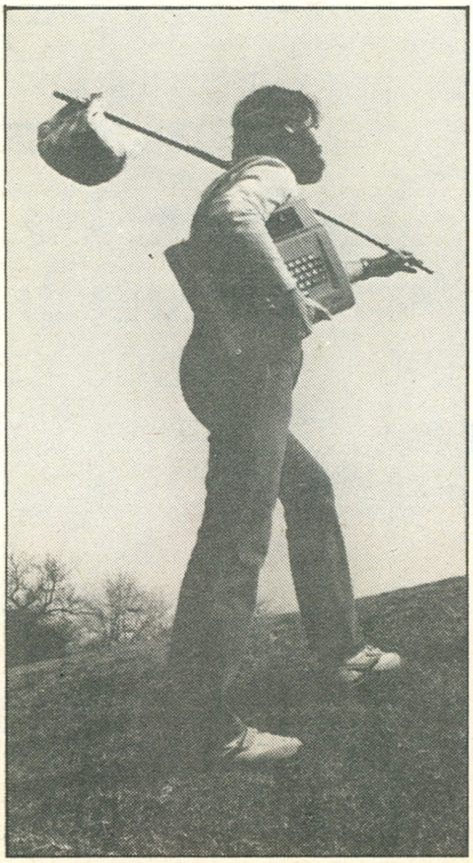
Memóriaméret: 16 kbyte ROM, 17 kbyte RAM.

Súly: 2,6 kg.

Méret: 38x30x8 cm.

Képernyő: az első szériák csak saját tv-vel, a későbbiek „háztartási" tv-vel működnek, fekete-fehér képpel, képernyőátmérő 31 cm, 24x48-as.

Csatlakozás: eredetileg egy, később két speciális magnetofonnal együtt szállították.





ságot: az ABC-tulajdonosok többségénél csak magnetofonnal használják.

Képernyőkezelés – 3,2

3 KÍN

Hát ez sem egy lenyűgöző osztályzat. A gép fekete-fehérben dolgozik, durva grafikával, ez akkoriban elegendőnek tűnt. A továbbiakban előnyének és hátrányának ugyanazokat a dolgokat nevezték meg inkvizítoraink: van, aki könnyen kezelhetőnek, van, aki túl bonyolultnak tartja, az egyik felhasználó kedveli, hogy minden karakternél meg kell adni, hogy alfanumerikus vagy grafikus karaktert használunk-e, mások ezt igen nehézkesnek tartják. Feltétlenül előnyként fogalmazta mindenki, hogy a gép eredeti karakterkészlete tartalmaz ékezetes betűket is. Ez ugyan sajátosan magyar igény, de ez a gép ezt is teljesíti.

Hang – 3,5

4 KÍN

A gép egy tiszta hang- egy zaj- és egy impulzusgenerátorral rendelkezik, amik BASIC-ből elérhetőek. Fiatalabb inkvizítoraink, akik már a több szólamú, zenélő csodákhoz vannak szokva, úgy érezték, hogy ez kevés. Abban azonban mindenki egyetértett, hogy a hang könnyen elérhető, egyszerűen programozható és szép. Van egy nagy előnye, ami a modern új gépeken már sajnos nem így van (Commodore, HT), hogy egyszerű hangot egyszerű programmal lehet készíteni, bonyolultat – bonyolultabb programmal. Az „új" gépeknél több sor és tizenegynéhány adat kell, hogy a legegyszerűbb fütty megszólaljon. Úgy látszik, ezt azóta elfelejtették a gépek.

Kazettás tárolás megbízhatósága – 3,4

5 KÍN

„Amit kétszer felvettem, az soha nem veszett el, amit csak egyszer – az mindig!" A megbízhatósággal tehát nem teljesen elégedettek inkvizítoraink, másik magnó használata esetén bizonytalan a betöltés, az adatokat kivinni könnyebb, mint később megtalálni. A téma külön érdekessége, hogy a kezdeti konstrukciót módosítva később a BRG két magnetofonnal szállította a gépet. Mindkét magnó gyakorlatilag ugyanarra a funkcióra készült, így a rosszmájúak szerint csak annyi az előnye, hogy a kettő közül talán egy működik is.

Gépi kódú programozás – 3,2

6 KÍN

A gépen nincs monitor, a gépi kódú program csak bűvészkedéssel SAVE-elhető, tehát kissé bonyolult, de ha valaki megszokta, akkor mégis kényelmesnek mondja.

Megbízhatóság – 4,5

7 KÍN

Ebben a kínban mindig az alapgép megbízhatóságát osztályozzuk, amennyire ez persze függetleníthető. A gép életkorához képest meglepően szép eredmény, gyakorlatilag az alapgéppel senkinek nem volt baja. Aki gyengébb osztályzatot adott, az általában a két magnetofonos kivitelt ismeri, amelyről köztudott, hogy a magnetofonok tápegysége fűti a gépet is. Az ABC-hívők a jó osztályzat mellé azt is hozzátették, szerintük a gép túl fog élni minket is, van olyan, amelyik hároméves használat alatt egyszer sem romlott el.

Billentyűzet – 4,9

8 KÍN

Az osztályzat magáért beszél: szinte tökéletes, nem nagy és nem kicsi, kényelmes, pont akkor ismétel, amikor kell és ismét elhangzott, hogy azon kevés gépek közé tartozik, amely ismeri az ékezetes betűket. Azt persze tudjuk, hogy ezt nem a kedvünkért fejlesztették így, hanem a svéd ábécé kedvéért, de mégis jólesik.

Dokumentáció – 3,2

9 KÍN

Eredeti gépkönyvvel nem sokan találkoztak. A géphez itthon adott változat állítólag az eredeti szűkített fordítása, amelyről nem túl jó vélemény alakult ki. Tankönyvként nemigen használható, gépkönyvnek viszont nem elég részletes és helyenként pontatlan is.

Editálás – 3,9

10 KÍN

Példásan egyszerű a javítás, ez az előnye és persze a hátránya is. Gyors, könnyen elérhető, de hosszú soroknál mégis lassúvá válik. Ha egy hosszú sorban egy karakter hibás, akkor a gép törli az egész sort, lehet kezdeni elölről. Ez nem kimondott előny. Jó a sorszámjavítási lehetőség, és van néhány olyan funkció, ami a modernebb nagyobb gépeken sincs (TRACE, ON ERROR GOTO). Több sort viszont nem lehet törölni egyszerre.

A gép programnyelve – 4,4

11 KÍN

Inkvizítoraink szinte kivétel nélkül megegyeztek abban, hogy az alap BASIC minden lényeges tartalmaz, ami elvárható. És abban is, hogy semmi különleges nincs benne. Ami mégis kimagasló, az a sebessége, máig az egyik leggyorsabb gépnek számít. Valaki úgy fogalmazott, hogy érezhető, hogy ez „európai" szoftver, mert az amerikai fejlesztésű gépeknél inkább a hardverre figyelnek oda.

Tanulhatóság – 4,8

12 KÍN

Az osztályzathoz sok kommentár nem szükséges, oktatógépnek készült, és mint látható, erre kiválóan alkalmas. Külön előnyként emlí-

VARGA ANTAL: Diákjaim gyorsabban megtanulják programjaikat javítani mint megírni, olyan jó a gép editálása!

ÚJVÁRI ÉVA: Kezdők már az első rövid magyarázat után is tudnak rajta programozni!

LIPOVSZKI GÁBOR: Nagyon szeretem!

TÖRÖK TURUL: Egy régi fiú a PC-k őskorából, de sokunk számára ő volt az „Első"

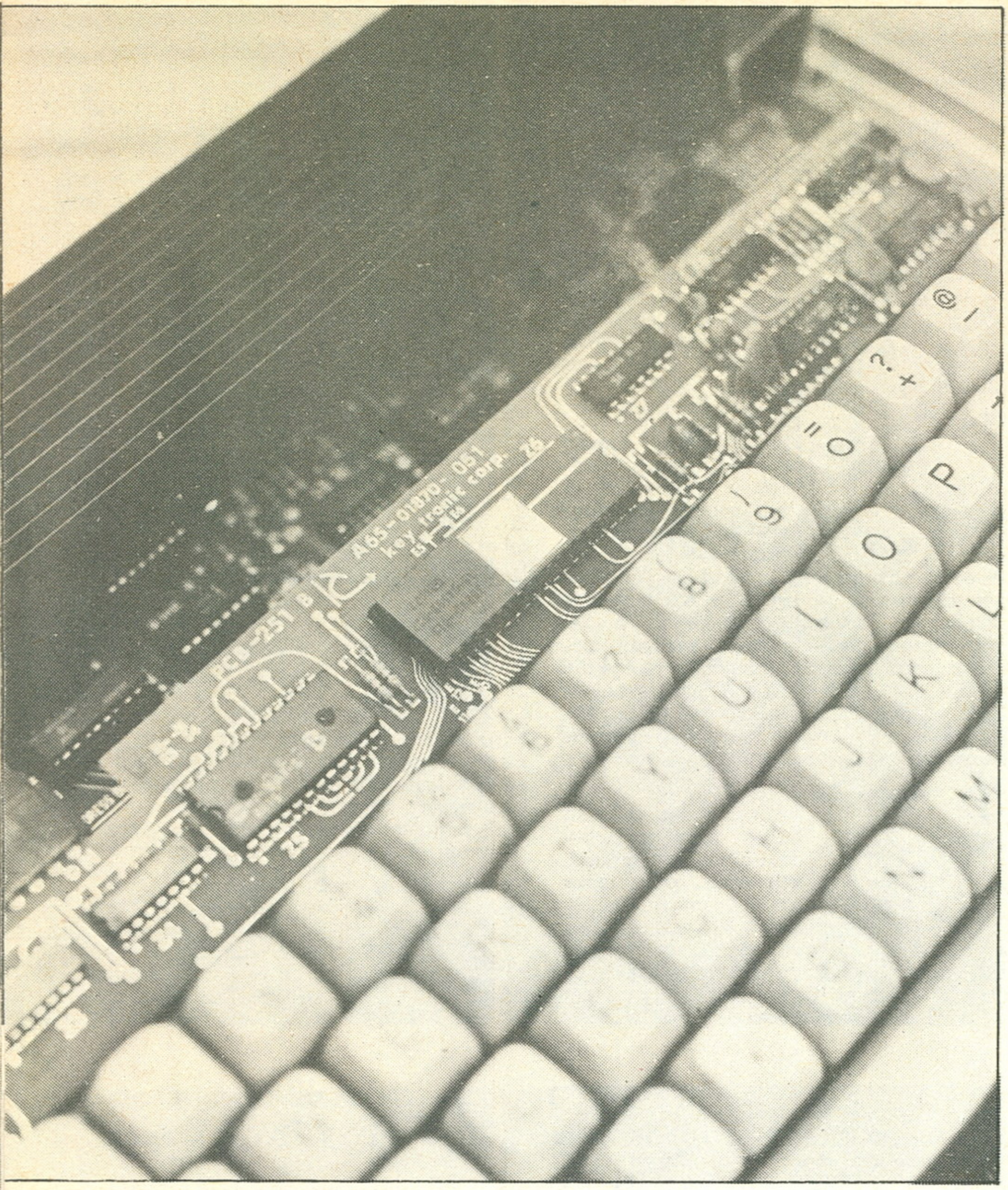
HALÁSZ PÉTER: Néhány olyan dolgot nem tud, amit ma már egy hasonló kategóriájú géptől elvárhatunk!

KŐRÖSI ÁKOS: Könnyen bele lehet szeretni, de néha csalódik benne az ember!



SZUPER

KÍNPADON A ABC-80



tette valaki – ami ma már nem ritka, de mégsem általános –, hogy a gép bekapcsolás után azonnal működik. Tehát nem kell különféle mutatványokat végezni ahhoz, hogy használni lehessen (lásd még HT).

Emberközelség – 4,4

Talán ebben a témában nem kell elfogultsággal vádolni inkvizítorainkat – valóban barátságos gépnek tűnik. Bár itt is felszínre került a nemzedéki ellentét: a fiatalabbak csak-csak több hibát találtak és más gépekkel is jobban megbarátkoztak.

Szubjektív vélemény – 4,3

Ahhoz képest, hogy ma már a szakma öregjének számít a gép, nem is rossz az osztályzat. És talán ebből érthető meg a legjobban a siker: a hetvenes évek végén kifejlesztett gép már akkor a nyolcvanas évekre készült. Okosan tervezett gyors, kézhezálló konstrukció, ami a mai számítástechnikai szakemberek első mikrogépe volt – érthető, hogy megszerették. Előrelátóan készült a gép, mert – bár akkor még nem volt divat, mégis tud floppyt is kezelni és – egyetlen inkvizítorunk tapasztalatai szerint – kiválóan. A jó osztályzatokat és a könnybe lábadó szemeket látva persze azonnal felmerül a kérdés: akkor miért nem gyártják, hova tűnt? Bár erről pontos, a gyártól származó információink nincsenek, megpróbáltuk kitalálni az okát. Úgy érezzük, hogy a svéd gép nem bírta a világpiac próbáját, nem tudta felvenni a harcot a tömeggyártású amerikai, japán, angol gépekkel, így elsősorban az ára miatt lemaradt. Ami azonban a kihalás talán közvetlen kiváló oka lehetett, az más, az értékesítési esélyek változása. Magát a gépet előrelátóan tervezték, azt azonban nem ismerték fel, hogy a jövő a programgyártás. Ma már szinte csak olyan gépeket lehet eladni a piacon, amihez nagy mennyiségű program, programcsomag is kapható. A fogyasztók tömege, a háziasszonyok és farmerek nem akarnak programozni tanulni – csak használni szeretnék a gépet, és ezt az ABC 80 már nem bírta. Még létezik, de már nincs – a szakemberek tartsák meg emlékezetükben!

ABC 80
VALLATASANAK EREDMENYE

KINOK	HALASZ PETER EGYETEMI HALLGATO 1	BRANYI LASZLO FOISK. HALLGATO 2	TOROK TURUL MATEMATIKUS 3	UJVAKI EVA EGYETEMI HALLGATO 4	ZSAKO LASZLO MATEMATIKUS 5	VARGA ANTAL KOZEPISK. TANAR 6	KOROSI AKOS KOZEPISK. TANULO 7	LIPOVSZKI GABOR PROGRAMOZO 8	ATLAG
1 .KIN:AR	–	–	–	–	–	–	–	–	–
2 .KIN:PERIFERIAK	3/4	3	3	–	4	3	2/3	3	3.1
3 .KIN:KEPERNYOKEZELES	3/4	3	2/3	3	3	3	3/4	4	3.2
4 .KIN:HANG	3	3	3/4	4	4	4	3/4	3	3.5
5 .KIN:KAZETTAS TAROLAS	3	3	3	4	4	3/4	3	4	3.4
6 .KIN:GEPI KODU PROGRAMOZAS	3	3	3	–	4	3	–	3	3.2
7 .KIN:MEGBIZHATOSAG	4/5	4	4/5	5	5	4	4	5	4.5
8 .KIN:BILLENTYUZET	5	4	5	5	5	5	5	5	4.9
9 .KIN:DOKUMENTACIO	3/4	2/3	4	3	3	3	3/4	3	3.2
10 .KIN:EDITALAS	3	3	4/5	4	4	4/5	4/5	4	3.9
11 .KIN:A GEP PROGRAMNYELVE	4/5	3/4	4/5	5	5	4	4	5	4.4
12 .KIN:TANULHATOSAG	4	4/5	4/5	5	5	5	5	5	4.8
13 .KIN:EMBERKOZELSEG	4	4	4	5	5	4	4	5	4.4
+ 1 KIN:SZUBJEKTIV VELEMENY	3/4	3/4	4	5	5	4	4	5	4.3
ATLAG	3.7	3.4	3.8	4.4	4.3	3.8	3.9	4.2	3.9

NOVOTRADE RT

HAJLÉKONY MÁGNESLEMEZES TÁROLÓ SFD 1001 1 MBYTE

Bérleti díja: 230 000 Ft 3 évre

Az SFD intelligens tároló periféria a COMMODORE számítógéphez. Ezt a berendezést külön is figyelmébe ajánljuk a COMMODORE 64 típusú gépeket már alkalmazó felhasználóknak.
Az SFD 1001 FLOPPY diszk használatával a számítógép ***háttértárolási kapacitása*** a hagyományos VC 1541-es floppy diszkhez viszonyítva ***6–7-szeresére nő,*** s nem elhanyagolható, hogy az ÁTVITELI SEBESSÉGE is közel ÖTSZÖRÖS. Ennek ismeretében tehát a C–64-es számítógép felhasználásának új területei nyílnak meg.
Az SFD 1001-es FLOPPY közvetett kapcsolatot teremt a C–64-es a C–64 és a C–610-es gépek között is, hiszen mindkettőhöz hozzákapcsolva egy-egy egységet, biztosítható az adatforgalom mindkét irányban.
Az IEEE 488 felülettel rendelkező COMMODORE 500, 600 és 700-as sorozatú számítógépekhez is közvetlenül kapcsolható.
A COMMODORE 64 típusú gépekhez az általunk szállított csatlakozóval a felhasználó állíthatja össze a konfigurációt, amely azonnal működőképes.
A DOS parancskészlete a számítógép BASIC utasításrendszerével magas szintű file kezelési lehetőségeket biztosít.
IEE 488-as illesztő áramkör C–64-hez: 15 900,– Ft.
Szállítási szerződés azonnal!

FŐKÖNYVI ÉS FOLYÓSZÁMLA-KÖNYVELÉS C 64-RE – VC 1541 – ÉS SFD 1001-ES FLOPPYRA – VALAMINT C 610-RE.

A programcsomag a gazdálkodó egység ***eszköz- és forrásállományában bekövetkezett változásokról ad idősoros és számlasoros elszámolást.*** Együttesen kezeli a főkönyvi és a folyószámla-könyvelést, biztosítva ezáltal a számviteli előírásoknak megfelelő egyezőségeket. Lehetőség van arra is, hogy a használó a programcsomagot kizárólag a főkönyvi könyvelésre vagy csak a folyószámla-könyvelésre használja.
A programcsomag a gazdasági műveletek idősoros elszámolására naplót készít:

- a beérkezett számlákról,
- a kiment számlákról,
- a banki forgalomról,
- a pénztárak forgalmáról,
- a vegyes tételekről.

A gazdasági műveletek tételei a naplóba gyűjtéssel egyidejűleg a főkönyvi és/vagy a folyószámlára kerülnek. A költségeket elsődlegesen a költséghelyszámlákra könyveli, úgy hogy automatikusan a költségnemszámlára gyűjti őket.
A bejött és a kiment számlák könyvelése változó ellenszámlákkal, vagy fixen rögzített naplók gyűjtésének megfelelő ellenszámlákkal történhet.
Készítette: ISTAR GM

ISES
STATISZTIKAI KIÉRTÉKELŐ PROGRAMRENDSZER C–64-RE

Az ISES számos tudományos kutatás adatainak matematikai statisztikai kiértékelése során leszűrődött tapasztalatokat foglalja egységes zárt rendszerbe úgy, hogy a számítógépes ismeretekkel nem rendelkező kutató is nehézség nélkül elvégezheti adatainak feldolgozását.

- Felxibilis adatbevitel, -módosítás, -ellenőrzés
- Sokféle táblázási és hisztogramkészítési lehetőség (több dimenzióban is)
- Új változók, indexek képzése, elemzése
- Statisztikai próbák, normalitásvizsgálat, varianciaanalízis
- Többféle kísérleti elrendezés esetére
- Faktoranalízis, többféle klaszteranalízis, főkomponensanalízis, kovarianciaanalízis
- Item-analízis, Rasch-modell
- Korreláció-, regresszió- és trendanalízis

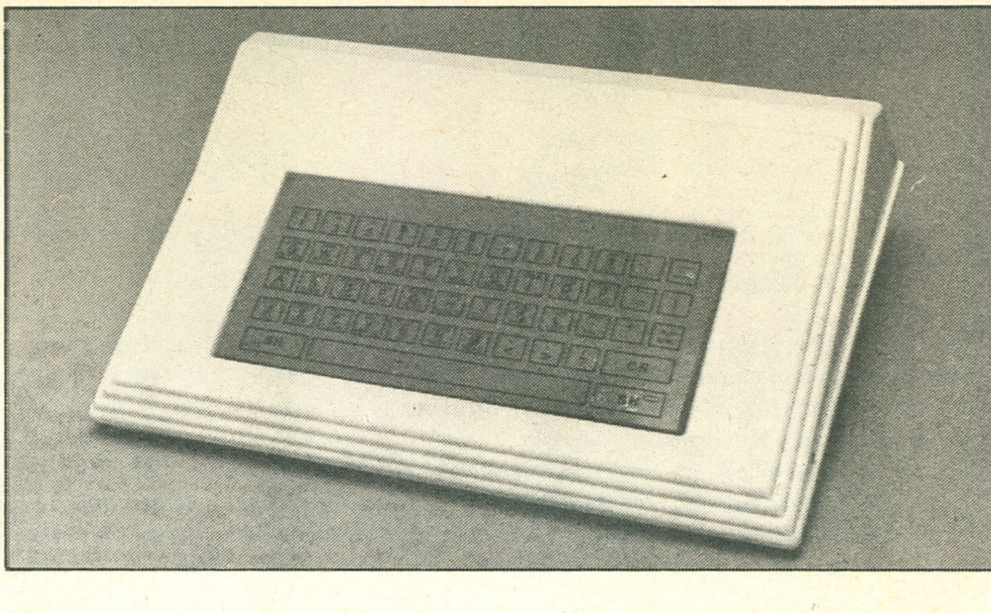
A 3 lemezt megtöltő programok használatát részletes felhasználói kézikönyv segíti, amely tartalmazza számos sztenderd vizsgálati módszer elvégzésének „kottáját" és ötleteket ad a rendszer bonyolultabb, egyedi problémákra való alkalmazásához.

Az ISES rendszer ára (3 lemez + kézikönyv): 37 500 Ft.
NOVOTRADE Rt.
Budapest XIII., Kresz Géza u. 14. Telefon: 122-047, 122-095. Telex: 22-7673

SZUPER VALLATÓ

Egy olyan gép vallatását tűztük napirendre, amelyből néhány száz darab van mindössze az országban, mégis komoly jelentősége van a hazai számítástechnikában. Magyar Apple-nek is szokták hívni, utalva ezzel ama világhírű gépcsoda és az AIRCOMP születése közti hasonlóságra. Mindkettőt fiatal „srácok" hozták össze egy „sufniban" berendezett műhelyecskében, s ráadásul mindkettő tartalmaz egy sor olyan kreatív újítást, ami csak „kamaszkorú" zseniknek juthat eszébe, akiket nem kötnek még konvenciók, hagyományok. Az AIRCOMP feltehetőleg jobb sorsra érdemes konstrukció, mint ami lett belőle. A Lukács testvérek – ők azok a bizonyos „kamasz" konstruktőrök – végül is a BOSCOOP cégnél leltek „menedéket" támogatást. Csakhogy a BOSCOOP nem rendelkezett, egyelőre nem is rendelkezik olyan igazi háttérrel és tőkével, hogy az AIRCOMP-ot komolyabb szériában gyárthatná. Az eddig eladott százegynéhány gépen kívül több tízes nagyságrendű barkácsolt példány is van az országban belőle. Legfeljebb a neve egy kicsit más, nem AIRCOMP, hanem HOMELAB (Home = otthon, Lab = laboratórium). A HCC klub, azaz az amatőr számítógép-építők klubja országot átfogó hálózatának tagjai ugyanis ezt a gépet építették, építik szerte az országban.

Vallatásunkon kihasználva a lehetőséget, meghívottként részt vett a konstruktőr Lukács József – testvére, Endre „igazoltan" volt távol; katona –, valamint a gyártó PERSONAL Agroelektronikai GT elnöke, Cseres Pál; és számítástechnikai fejlesztőmérnöke, Diebel Dietrich.



GYÁRI ADATOK

Memóriaegység: 16 kbyte RAM
Csatlakozási lehetőségek: tv-készülék (fekete-fehér), magnetofon, monitor
Súly: alapgép: 0,9 kg, tápegység 1,2 kg
Méret: gép: 31x21x5 cm – tápegység 12x14x5 cm

KÍNRENDSZER

Kicsit kínban voltunk, amikor a Vallatásra egybegyűlt inkvizítorainkkal megbeszéltük az osztályozási alapelveket és a kínrendszert. Kiderült ugyanis, hogy a jelenlévők fele az AIRCOMP 16 fehér színű műanyag dobozban forgalomba hozott – barátilag szappanosdoboznak keresztelt – első szériát ismeri, míg a többiek a második, fekete dobozban világot látott szériát. Ráadásul az is kiderült a beszélgetésből, hogy a két széria minősége közt óriási a különbség bizonyos kínokban. Hosszas vita után úgy döntöttünk, hogy ennek ellenére együtt értékeljük a kettőnek nem mondható, de egynek sem egészen elfogadható gépet. Erre elsősorban az jogosított föl bennünket, hogy a gyártó és forgalmazó, a PERSONAL Agroelektronikai GT, illetve a BOSCOOP sem érezte szükségesnek, hogy bármilyen megkülönböztető jelzéssel – betűvel vagy perjellel és számmal – válassza el egymástól a két szériát. Ily módon az átlagosztályzatok egybemossák a két széria nívókülönbségét. Ezt azzal kívánjuk enyhíteni, hogy az egyes kínok kifejtésénél jelezzük, ha komolyabb eltérés van a két szériát osztályozók jegyei közt.

Ár – 4,2

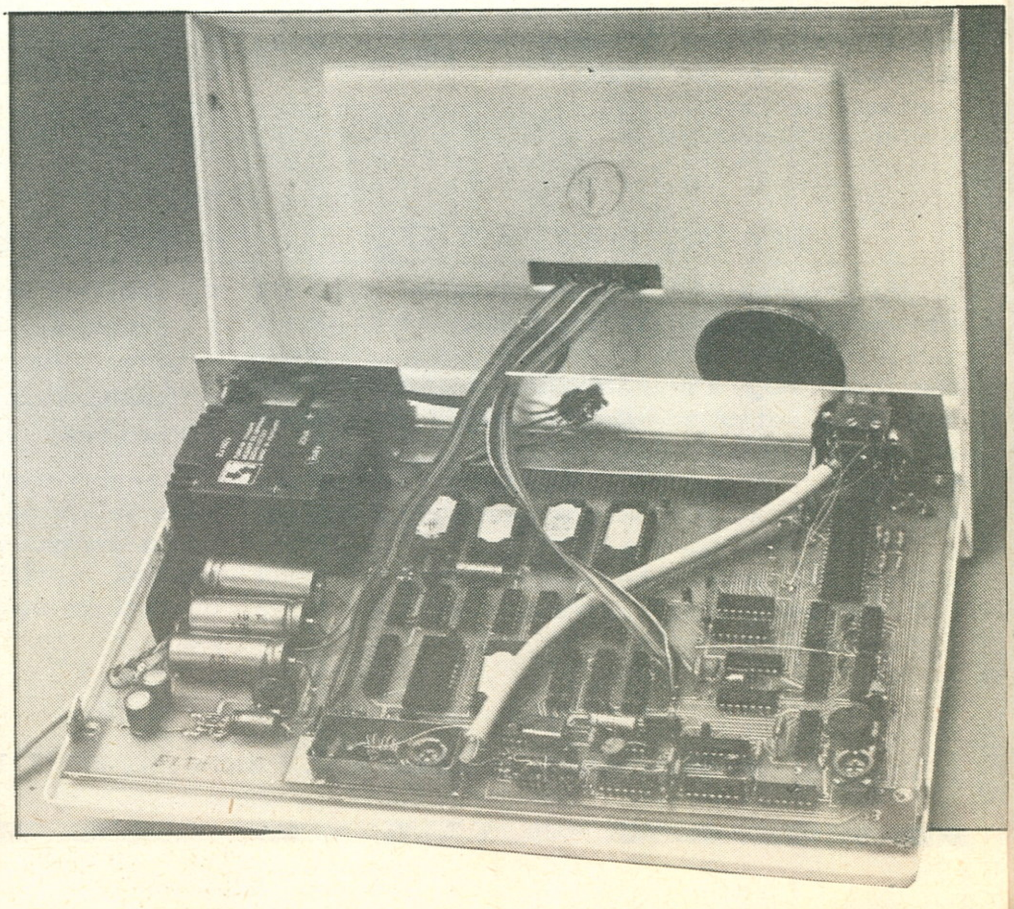
1 KÍN

Külföldi gépeknél a „hazai vagy a kinti" dilemmája szokásos nagy kérdőjel. Hazai gyártmányú gépet Vallatónkban másodszor vizsgáltunk. Mint a HT-nél, itt is az volt a nagy kérdésünk, hogy mihez képest? Egyik inkvizítorunk tréfásan azt mondta erre, hogy „A Vasedényhez képest". Valójában inkvizítoraink úgy érezték, hogy ez az ár induláskor forradalmian alacsony volt, ma már jó lenne, ha nem lenne, vagyis, hogy kevesebb lenne. A gyártó jelen lévő képviselője „megnyugtatott" bennünket, hogy árváltozás nem várható. Azt is őszintén elmondták, hogy kisebb tőkés hányaddal elő tudnák állítani a gépet, de ez nemhogy költségcsökkentést nem eredményezne, hanem ellenkezőleg, a rubelimport növelné a fogyasztói árat.

Perifériák – 2,2

2 KÍN

A két szélsőség ebben a kínban az elégtelen és a jó volt. Meglehetősen nagy különbség. Előbbi indoka rendkívül egyszerű; jelen pillanatban a géphez az égvilágon semmilyen plusz periféria sincs, a forgalomban lévők közt sem találunk olyat, amely egyszerű csatlakoztatással működtethető lenne. A négyest adó inkvizítor viszont mint felhasználó úgy érzi, hogy neki a magnetofonos tárolón és a tv-n kívül másra nincs és nem is lenne szüksége. Ő tudja! A tényszerű tájékoztatás megkívánja, hogy azt is leírjuk, e két fenti – a többség által alapnak és nem perifériának tekintett – megjelenítőn, illetve tárolón kívül monitort is csatlakoztathatunk a géphez. Egyik inkvizítorunk azzal indokolta az általa adott hármast, hogy: „még nincsenek sorozatban gyártott interface-ek és perifériák, a jegy egyben a bizakodást is kifejezi". A gyártó képviselője valóban adott némi alapot a bizakodásra, mert ígérete szerint nemsokára kapható lesz az Aircomphoz soros és párhuzamos interface, előkészületben van analóg/digitál átalakító és kétállapotú jeleket fogadó és kibocsátó egység.



KÍNPADON A AIRCOMP

Képernyőkezelés – 4,4

Amint az átlagosztályzat is mutatja, ezzel lényegében mindenki elégedett. A siker elsősorban a nagy felbontású grafikának (320x200 pont) köszönhető. Jó lehetőséget nyújt a programozónak még az is, hogy a képernyő tetszés szerinti arányban felosztható grafikus és szöveges területre. Hiányzik a TAB, a PRINT AT – bár mint az a konstruktőr Lukács József írásából kiderül, szubrutinként elérhető. Néhányan a színeket is hiányolták, bár erre könnyű szívvel mondhatjuk azt, hogy „No, de kérem, ennyiért?" Hasonlóképpen könnyű szívvel elsiklunk a nagy felbontású grafika lassúsága fölött. Már az is nagy szó, hogy egyáltalán van!

Hang – 3,3

Az Aircomp azon kevés gépek közé tartozik, amelyeknél kétféle hangról kell beszélni. Mindkettőt ugyanaz a hanggenerátor bocsátja ki, de más a funkciója, s más az elérhetősége. Az egyik az úgynevezett „segédhang". Ez minden billentyű megnyomásakor megszólal, mintegy hangban igazolja vissza, hogy a gép elfogadta a billentyű jelét. Ez roppant előnyös és szükséges segítség a programozáskor, használatkor. Hiszen az ilyen érintőfóliás tasztatúra hangsegítség nélkül olyan, mint a vak koldus bot nélkül. Ráadásul az Aricomp tervezői még arra is figyelemmel voltak, hogy a SHIFT billentyű egyidejű használatakor magasabb hangot halljunk, mint ugyanannak a billentyűnek SHIFT nélküli használatakor. (Azt már csak a „lelki csipkével" rendelkezők tudják igazán értékelni, hogy kétféle SHIFT kétféle hangmagasság-változást von maga után!) Ez a hang egyértelmű heurékát váltott ki inkvizítorainkból. A másik, az „igazi" hang, az, amely BASIC programozással elérhető és „muzsikálásra" ingerli a számítógépet, már kevesebb hívet szerzett Aircompos körökben. Nehézkes, ha megszólal, felfüggeszti a a képernyőkezelést, néhányan szívesen látnának egy külön hangszórókimenetet is. Őszintén szólva lapunk megérti azt a gondolatmenetet, amely a gyártótól származik, s amely röviden úgy foglalható össze: ennek a gépnek voltaképpen még ennyi hang is sok! A hang valóban nem tartozik az alapkövetelmények közé, jelzi ezt a szűk frekvenciatartomány is, amely szintén nem tetszett egynémely inkvizítorunknak.

Kazettás tárolás megbízhatósága – 3,7

Ez az első olyan kín a 14-ből, amelynél komoly eltérést mutat a régi „szappandobozt" értékelők és a fekete változatot ismerők és osztályozók véleménye. Érdekességként külön is átlagoltuk a két kategóriát. Nos a régi gép mindössze 2,8-et, az új 4,2-et ért el inkvizítorainknál. Mondhatnánk, kottázni lehet a különbséget. Az az érdekes, hogy a különbség mögött semmiféle konstrukciós változtatás sincs. Mindössze nagyobb gyártási tapasztalat és jobb hangolás. Érdekessége a dolognak, hogy Lukács József és Endre nem bonyolította túl a magnókiválasztást. Bementek az első Keravillba és megkérdezték, hogy melyik a legolcsóbb kazettás magnó. Az MK 27 – hangzott a válasz. – Nosza, akkor ehhez igazítjuk a gépet – döntöttek. Arról már sajnos nem ők tehetnek, hogy az MK 27 olyan „csodálatos" terméke a magyar iparnak, hogy ahány darab, annyiféle, s ráadásul még csak nem is szabványos a kimenőjel szintje, amiből mindössze annyi következik, hogy hiába van valakinek drága, jó minőségű magnója, lehet, hogy még annyira sem megy vele, mint az olcsó MK 27-tel. Inkvizítoraink közt egyébként volt, aki maximálisan elégedett a kazettás tárolás, ki- és beolvasás biztonságával, mások átlagosnak mondták, megint mások a jó minőségű kazetta fontosságára hívták fel a figyelmet, csakúgy, mint más számítógépeknél. Mindenesetre úgy tűnik, hogy a szappandoboz egyértelműen rossz színvonalú be- és kiolvasását megbízható szintre emelték a gyártók.

Gépi kódú programozás – 4,7

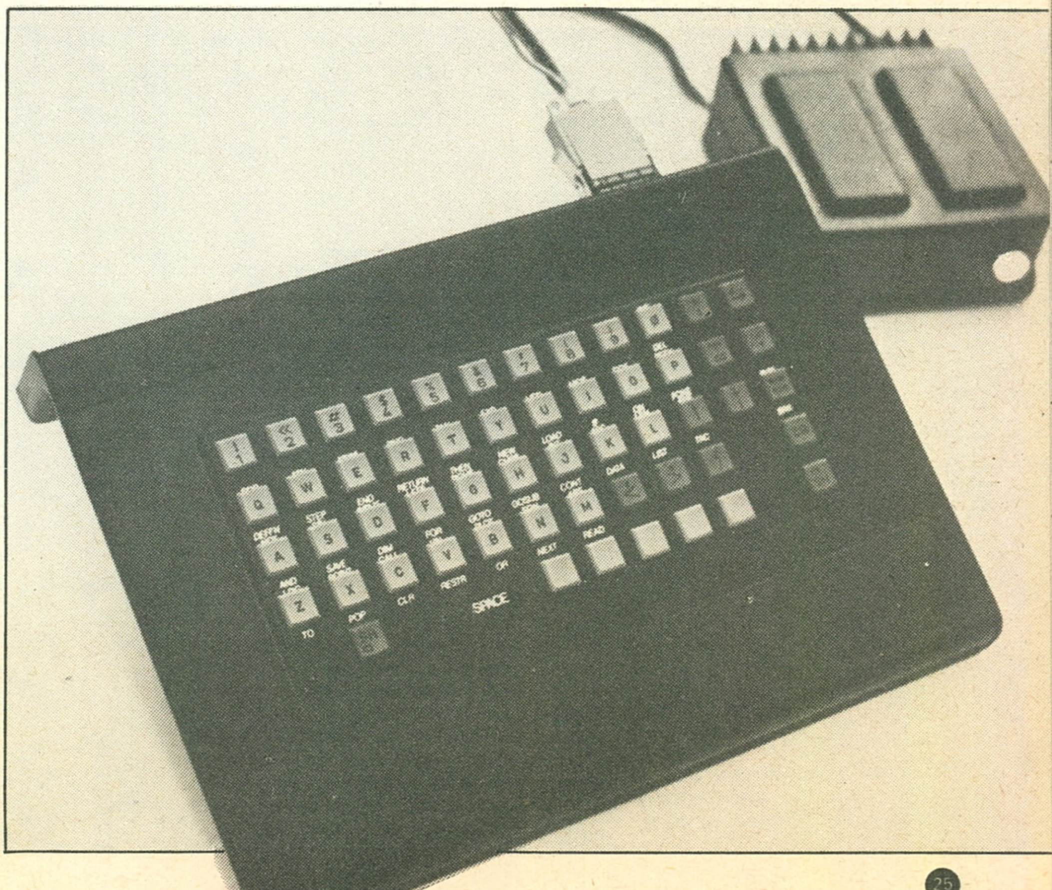
Táblázatunkból is kitűnik, hogy ez a legmagasabb, legegyenletesebb osztályzatsor. Átlaga is közelebb jár a tökéleteshez. Gépi kódú programozásban valóban lényegesen többet tud ez a gép, mint hasonló kategóriájú társai szerte a világban. Van monitor üzemmódja, amelyben lehetséges programot bevinni, kilistázni, futtatni, kazettára vinni és kazettára kivitt programot ráadásul úgy lehet beolvasni, akárha BASIC programot töltenénk be. Egyedülálló lehetőségeket nyújt a géphez vásárolható két különböző EPROM: ASSEMBLER (6500 forint), és a DEBUGGER (4500 forint). Mindezek után adódik a kérdés, hogy vajon miért nem lett színjeles e kín osztályzata? Nos, éppen az ASSEMBLER-t hiányolták néhányan, örömmel értesültek róla, hogy nyílt kapukat döngetnek. Lehet, hogy ma már ők is jelest adnának?

Megbízhatóság – 3,8

Ismét egy olyan pont, amelynél érdemes kettéválasztani a két „generációt". A kettő közti különbség talán itt a legkiugróbb. Jó példa erre az az inkvizítorunk, aki mivel mindkét szériát ismeri, mindkettőt le is osztályozta. S míg a régebbi számára egyszerűen 1-est adott, addig az újabbra 4-est. A két széria átlaga is ég és föld: fehér színű szappandoboz: 2,3 – fekete színű új széria: 4,6!!! Mondjuk ki tehát feketén (új) fehéren (régi), hogy az a bizonyos szappandoboz egy elhamarkodott nullszéria volt. Aminthogy Vallatásunkon a gyártó cég képviselője önkritikusan be is ismerte, ha okosan és taktikusan jártak volna el, valóban nullszériának tekintették volna, s nem pénzért, hanem próbagépként, tesztelésre adták volna ki. (Szerkesztői megjegyzésként kikívánkozik belőlünk – a gyártó ugyan nem erősítette meg –, hogy a történtek nemcsak a gyártói, menedzselési rutin hiányára vezethetők vissza, hanem a tőkeerő hiányára és a pénzügyi szabályozók hibáira is.)

Billentyűk – 3,1

Az osztályzatok közül megint kilóg néhány kettes. Ismét a szappandoboz, amely éppen csak hogy nem lett 1-es. Kicsik a betűfelületek, lekoptak, nem lehet érezni a betűmezők határát. A fekete gépnél sem éppen rózsás a helyzet. Bár az érintőfóliás megoldás talán tovább ennél nem is fejleszthető. Ez a tasztatúra most már nem igényel komolyabb nyomást, megbízhatóan reagál, méretei is kényelmesek, kezelhetőek, ráadásul szépen muzsikál is a gép az egyes nyomások visszajelzésére. Ezzel együtt az érintőfólia mégis csak érintőfólia marad, és valljuk meg: „normális programozó" még ha tudja is, hogy ez komoly megtakarítást jelent, s a gép árát lényegesen csökkenti, mégsem képes túlságosan magas osztályzatot adni ilyen tasztatúrával rendelkező gépnek. Ami viszont az utóbbi hónapok azon pletykáit illeti, hogy „itt" a billentyűs Aircomp!" sajnos a gyártó képviselőitől megtudtuk, hogy az „itt" egyelőre túlzás. Alig néhány ilyen billentyűs gépet készítettek eddig, s beszerzési, valamint árproblémák miatt gyártása nem is várható.



SZUPER VALLATÓ

Dokumentáció – 2,4

9 KÍN

Igazság szerint ez érdemelné a legrosszabb átlagosztályzatot. Hogy egy tizeddel mégis a második helyre került (természetesen hátulról) a kínok rangsorában, az csak annak az inkvizítorunknak köszönhető, aki lévén közel a „tűzhöz," azaz a gyártóhoz, már jól ismerte a Vallatás időpontjában is az azóta remélhetőleg valamennyi géptulajdonosnak megküldött új gépkönyvet. A többiek azonban csak a régit olvashatták, amelyről sajnos a legtöbb jó, amit elmondhatunk, egyik inkvizítorunk szavaival: „El lehet kezdeni vele dolgozni, és már ez is valami!" A gyártó képviselői elmondták, hogy bizony egy gépkönyv elkészítése, menedzselése majdnem olyan nehéz, mint gépet csinálni. Lukács Józsefnek többen nekiszegezték a kérdést, vajon miért nem írták meg ők a könyvet, mégiscsak ők ismerik legjobban a saját gépüket. Lukács József nagyon is indokoltan elhárította a kérdést, mondván a gyártó is többször próbálta őket rávenni erre, sikertelenül. Ők ugyanis úgy érzik, hogy az külön vénát kíván. Hogy az új, már tankönyvnek használható gépkönyv mennyire egyesíti ezeket az erényeket, kiderül a következő hónapokban.

Editálás – 4,4

10 KÍN

A magas osztályzatok elsősorban a javítás egyszerűségének köszönhetők, valamint annak a nem mindennapi ténynek, hogy ez a gép monitor üzemmódban is rendelkezik FULL SCREEN EDITOR lehetőséggel, s ez bizony ebben a kategóriában szinte egyedülálló. Különösen ZX-en nevelkedett programozók ennek a gépnek is felróják, hogy nem soronként vizsgálja a szintaktikus hibákat, azaz hibás sorokat is elfogad. Sajnos nincs sortörlési lehetőség sorszámtól sorszámig, valamint hiányzik az újrasorszámozás is.

A gép programnyelve – 3,8

11 KÍN

Vallatásunknak ezen a pontján fordult elő először és utoljára, hogy a gép jelenlevő konstruktőre, Lukács József addigi – hűvös nyugalmát félretéve szinte elsírta magát. – Elszorul a szívem ezeket az osztályzatokat hallva – mondta, majd így folytatta: – ez az a kín, amelyre legalább 5,2-et kellett volna kapnia a gépnek. Ha valami az AIRCOMP erőssége, akkor az épp a BASIC-je. A konstruktőrnek több dologban igaza volt. S ezt nem a szerkesztő mondja, így utólag, hanem az inkvizítorok mondták tizenöt perccel később. Kiderült ugyanis, hogy ők csak azt osztályozhatták, amit ismertek. Márpedig az AIRCOMP sok titkát ez ideig nem árulta el fölhasználóinak. Nagyon szemléletesen úgy fogalmazott az egyik inkvizítor, hogy: „hiába tud egy autó 240 km/órát, ha az órájában csak 120 van. Ehelyett inkább írjanak bele 300 km-t, s majd megyek vele, amennyit tudok". Azaz nem kellett volna titokban tartani a gép BASIC-jének egy sor jó tulajdonságát. Ez ismét a gépkönyv, a dokumentáció elégtelenségére vezethető vissza. Ezzel együtt néhány szellemes megoldást, finomságot felfedeztek inkvizítoraink. Valamennyien dicsérték például az IF szellemes kidolgozását, a POP utasítást vagy az ON zseniális megoldását.

Amit viszont hiányoltak inkvizítoraink: a már említett TAB-on DELETE sorszámtól sorszámig-on kívül, az INKEY, a RENUMBER, a MERGE, a VERIFY, a PRINT USING és az ON ERROR GOTO. Használhatatlannak tartják a DEFFN utasítást, s a különleges gyorsaság árakként felrójják a számítások pontatlanságát is.

Tanulhatóság – 4,3

12 KÍN

Úgy tűnik, ha a gép dokumentációja jobb lenne, s a kazettás tárolás megbízhatósága iránt semmi kétségük sem lenne a felhasználóknak, igen jól tanulható gép lenne az AIRCOMP. (Persze, ha a nagyapám az unokám lenne...) Inkvizítoraink külön erényének tartották, hogy a kulcsszavak külön billentyűkről is, de beírással is bevihetők. Nagyon nagy előnynek tartották a tanulhatóság szempontjából azt, hogy az editálás megengedi, sőt elvárja a próbálgatást. S ily módon a legjobban tanulható is.

Emberközelség – 4,3

13 KÍN

Azon kevesek, akik hozzáfértek már ehhez a géphez, kedvelik. Egyetlen, nemcsak erre a gépre, hanem minden hazai gyártmányú számítógépre érvényes megállapítás hangzott el a vallatáson. „Mi lenne, ha egy magyar gép magyarul is tudna? Például a hibaüzeneteket az angol szöveg mellett vagy helyett magyarul is közölné?" érdekes, megszívlelhető vélemény.

Szubjektív vélemény – 4,2

+1 KÍN

A magas átlagosztályzat csak alátámasztja az előző kínban írottakat. Az érintőbillentyűzet itt is ront néhány tizedet. Annak ellenére, hogy nem kifejezetten erre a célra szánta a gépet sem a konstruktőr, sem a gyártó, oktatási célra is kiválóan alkalmasnak találták inkvizítoraink. Egyikük sommásan így fogalmazta meg véleményét: „Jó, hogy van ez a gép!"

Amikor inkvizítoraink a Vallatás végén még egyszer szemügyre vették a teljes értékelést, némi lelkiismeretfurdalással konstatálták, hogy talán szigorúbbak voltak a kelleténél. Egyikük még azt is megjegyezte, hogy talán jobb lett volna, ha ezúttal nem is osztályozunk. Hiszen az írni, olvasni tanuló elsősöket sem osztályozzák az első félévben. Azután egy másik inkvizítor gyorsan hozzátette, lehet, hogy épp ez a megfelelő „pedagógiai fogás", hogy tudniillik nemcsak osztályoztak, de még szigorúak is voltak, mert így tudják valamenyi felhasználó érdekében „rugdosni a gyár-

KEMÉNY ENDRE: Kegyetlenül bosszant a periféria-, szoftverhiány

TICK JÓZSEF: A magánfelhasználó szempontjából drága!

KIRÁLY ZOLTÁN: Különösen jó tulajdonsága a gépnek a rendkívüli gyorsaság

FÖLDI PÉTER: Nálam bevágódott az AIRCOMP!

FODOR JÓZSEF: A gép oktatásra is kitűnő! Az „együtt eltöltött" idő alatt nagyon megszerettem!

BARABÁSI REZSŐ: Az ember-gép kapcsolat szempontjából igazán ideális



tót". A tréfás megfogalmazás mögött ki nem mondott elismerés volt. Elismerése annak, hogy a gép végül is néhány lelkes ember erőfeszítéseként, „sokak támogatása ellenére" jött létre. Elismerése annak, hogy ezt a gépet a gyártók voltaképpen egy meglehetősen jól behatárolt szakterületre, a mezőgazdaság számítástechnikai ellátására „hozták tető alá", de végül is olyan sikert értek el vele, hogy ma, már ha bírnák gyártási kapacitással, alkatrésszel, jóval szélesebb érdeklődésre tarthatnának számot. Ma még nyitott kérdés, hogy mi az AIRCOMP jövője. (A Szuper BIT-LET megjelenésekor már „csukott kérdés". Tudniillik a gép gyártása azóta megszűnt, gyártóját fölszámolták.) Magának a gépnek mint konstrukciónak, mint számítógépnek lenne jövője, s nemcsak a mezőgazdaságban. Különösen, ha megfelelő szoftverbázis is épülne köré, s ha „megfelelő helyen" is látnának benne annyi fantáziát, mint azokon a helyeken, ahol az eddig elkészült gépek működnek!



Önvédelem

Látva az inkvizítorok lesújtó véleményét az AIRCOMP BASIC-ről, úgy gondoltam, fel kell emelnem szavamat mellette. Bár mint fejlesztő, talán nem tudok kellőképpen elfogulatlan lenni, mégis az a véleményem, ez az interpreter egy nagyon jól használható darab. Azok az extra újdonságok, amelyekkel a BASIC rendelkezik, mindenképpen ezt támasztják alá. Sajnos a kínvallatás során kiderült, hogy még a rendszeres felhasználók sem eléggé ismerik ezeket. Felhasználva az újság adta nyilvánosságot, hadd járuljak hozzá én is egy-két ötlettel a gép hatékonyabb felhasználásához.

Két speciális karakter

a PRINT CHR$ (5) a CR változó „színére" törli a grafikus képet (ugyanez a CALL 7535-tel is elérhető).

A PRINT CHR$ (12) letörli az alfás képernyőt (ugyanez SHIFT/CR-rel is elérhető, illetve az is beépíthető egy stringbe).

Kiszámított GOTO, GOSUB, CALL

*Az utasítások után aritmetikai kifejezés is állhat, de ezek nem kezdődhetnek számmal. A számmal kezdődő kifejezést csak sorszámnak veszi, pl. GOTO A * 10 és nem GOTO 10 * A.*

PEEK, POKE

A 8000 alatti címek nem memóriához fordulnak, hanem I/O címeket szólítanak meg. Így I/O eszközök olvashatók, illetve írhatók.

PRINT AT helyett

*A CURSOR POINTER HEX 4014–4015 címen van. Ahova ez mutat, ott villog majd a cursor, illetve oda ír a gép. A következő programrészlet (praktikusan szubrutin) az Y sor X pozíciójába állítja a cursort: A = X+Y*40+1:B= INT (A/256): POKE 16404, A–B*256, B+192*

Gépi kódú tárolás – BASIC-ből

A következő gépi kódú program egy BASIC-CALL-lal meghíva kitárolja a megadott területet:

```
LD HL,#Start
LD BC,#End
LD DE (név eleje)
CALL 5E8
RET
```

(Tehát HL-ben a kezdet, BC-ben a vég, DE pedig a név első byte-jára mutat. A név végén 22 vagy 60 (HEX) karakter áll. Ha a név nem kell, DE pl. Egy 60-ra mutasson.)

Gépi kódú program elhelyezése a BASIC-ben

Legegyszerűbb, ha a programot így kezdjük:

```
1 GOTO 10
2 ".............."
10 REM IDE JÖN A PROGRAM.
```

A második sorba egy csupa space-ből álló stringet írunk be, olyan hosszút, amilyen hosszú a gépi program lesz. Monitorba átmenve 40A6-tól megtaláljuk a második sort, és a space (HEX 20) helyére beírhatjuk a gépi programot. Sajnos a program így nem tartalmazhat 22 vagy 60 (HEX) byte-ot, és BASIC-be visszatérve a 2. sor nem javítható – nem is szabad rámenni a cursorral.

A változó tábla kezdetére a 4018–19, a végére pedig a 401C–1D printerek mutatnak. Ha ezt a területet kitároljuk (pl. a korábban említett módon), akkor ezeket az adatokat ugyanez a program beolvashatja. Ez így lényegesen gyorsabb, mint a PRINT és az INPUT.

A programok kicsit gyorsabbak lesznek, ha:

- *A sűrűn használt változókat a program elején definiáljuk (DIM-mel vagy pl. A = A-val);*
- *Ne használjunk számkonstansokat – ezeket valahol tegyük bele a változókba;*
- *A GOTO és GOSUB lehetőleg a program elején levő sorokat hívja.*

Lukács József

SZUPKAY ISTVÁN: Kár, hogy a hazai alkatrészproblémák miatt ez a gép nem futotta ki a formáját!

KOVÁCS LEVENTE: **A gép kellemesen „puha", kellőképpen „lezser", ugyanakkor kötött, precíz!**

FÖLDI LÁSZLÓ: A billentyűzetet kényelmetlennek és fárasztónak tartom!

NÁVAY SÁNDOR: Napi 300 kilométereket utaztatom a gépet, de ezt is jól bírja. Megbízhatósága kiváló!



Finomságok

Olvastam az Aircomp tesztelését. Meglepett a BASIC-re adott gyenge osztályzat. Egy éven át a „szappantartón" is programozok és – gépkönyv hiányában – magam is találtam több „finomságot" a nyelvben. Remélem, az új gépkönyvben felfedik a lehetőségeket azok számára, akiknek nincs idejük kitapasztalni ezeket. A gép – különösen BASIC-je miatt – többre hivatott a pár százas szériánál. A sok jó tulajdonság közül az egyik a stringkezelése, amit ilyen formában sem a HT–1080Z, sem a Commodore 64 nem tud. Ezek a gépek, ha pl. A$-be egy függvényt adunk be $(x^2+3x-2x^5)x$ megadása után VAL(A$)-re nem számolják ki a kifejezés értékét, így elég körülményes egy-egy függvény bevitele. Az AIRCOMP-nál egyszerűbb a dolog. Ha pl. egy egyenlet gyökeit akarjuk kiszámolni (Newton–Raphoson módszer), ez egyenlet és deriváltjának bevitelét egyszerű INPUT-tal megoldhatjuk:

```
10 INPUT „AZ EGYENLET"; A$
20 INPUT „A DERIVÁLT"; B$
30 INPUT „KEZDETI X ÉRTÉK"; X
40 X=X–(VAL(A$)/(VAL(B$))
50 IF ABS(VAL(A$)/VAL(B$))>1E–7
THEN GOTO 40
60 PRINT „AZ EGYIK GYÖK"; X
70 GOTO 30
```

*Ez a programocska természetesen a felhasználástól függően javítható, bővíthető. (A polinom gyökeit pl. harmadfokú egyenletnél a kezdeti X = 10000;0; –10000-nél megkapjuk. Célszerű a beíráskor pl. X ↑ 2 alak helyett az X*X-et használni, mert a gép negatív számokat nem tud hatványozni.)*

Szerintem ez egy nagyon jól használ-

AIRCOMP 16

VALLATASANAK EREDMENYE

K I N O K	KOVACS LEVENTE KOZGAZD. SZAKTANACSADO	SZUPKAY ISTVAN RENDSZERSZERVEZO	KIRALY ZOLTAN EGYETEMI HALLG.	TICK JOZSEF PROGRAMOZO	BARABASI REZSO RENDSZERSZERVEZO	NAVAY SANDOR MEZ.GAZD. MERNOK	FODOR JOZSEF ELEKTRONIKAI MUSZERESZ	KEMENY ENDRE KOZEPISKOLAI TANAR	FOLDI PETER KOZEPISKOLAS	FOLDI LASZLO KOZEPISKOLAS	ATLAG
	1	2	3	4	5	6	7	8	9	10	
1 .KIN:AR	5	5	4/5	3	4	5	5	4	3/4	3	4.2
2 .KIN:PERIFERIAK	3	2	1/2	3	3	4	2	1	1	1	2.2
3 .KIN:KEPERNYOKEZELES	5	3	4/5	5	5	-	4/5	4	4	4/5	4.4
4 .KIN:HANG	4	2	-	3	3/4	5	3	3	3	3	3.3
5 .KIN:KAZETTAS TAROLAS	4	3	2	4	4	5	5	3	3/4	-	3.7
6 .KIN:GEPI KODU PROGRAMOZAS	4	5	5	5	4/5	-	5	-	5	4	4.7
7 .KIN:MEGBIZHATOSAG	5	3	2	-	4	5	5	4	2/3	-	3.8
8 .KIN:BILLENTYUZET	3/4	3	2	4	3/4	4	3	4	2	2	3.1
9 .KIN:DOKUMENTACIO	2	2	2	-	2	5	2	1	3	3	2.4
10 .KIN:EDITALAS	4/5	4	4	5	5	-	5	4	4	4	4.4
11 .KIN:A GEP PROGRAMNYELVE	3/4	4	4	3•	4	-	3/4	4	4/5	4	3.8
12 .KIN:TANULHATOSAG	3/4	5	4	4	4	5	4	5	4	4	4.3
13 .KIN:EMBERKOZELSEG	4/5	4	3/4	4	5	5	5	4	4	4	4.3
+ 1 KIN:SZUBJEKTIV VELEMENY	5	4	4	4	4/5	5	4/5	3/4	4	3/4	4.2
ATLAG	4.0	3.5	3.3	3.9	4.0	4.8	4.0	3.4	3.4	3.3	3.8

ható interpreter, többet érdemel a gép. Ennyit az AIRCOMP védelmében. Ha van rá lehetőség, szívesen részt vennék a PRIMO tesztelésében. Remélem, az a gép is sok kellemes meglepetést tartogat.

Kovács Barna
egyetemi hallgató, Szeged

Mit tud Ön az ON-RÓL

Az AIRCOMP-16 BASIC-jének egyik legzseniálisabb „húzása": az ON UTASÍTÁS.

Az ON utasítás általában minden BASIC-ben megtalálható, a következő formában:

ON kifejezés GOTO 1. sorszám, 2. sorszám...

Ezt a gép úgy hajtja végre, hogy kiszámítja a kifejezés értékét, s ha van a GOTO után annyiadik sorszám, amennyi a kifejezés értékének egész része, akkor a sorszám által meghatározott sorra ugrik, különben a futás a következő soron folytatódik. GOTO helyett GOSUB is írható.

Az AIRCOMP-16 BASIC ON utasítása viszont a következő alakú:

ON kifejezés '1. utasításcsoport, '2. utasításcsoport, '...

ahol az utasításcsoportok állhatnak egy vagy több – kettősponttal elválasztott – utasításból. A hatás annyiban különbözik az előzőekben tárgyaltaktól, hogy a gép megkeresi az annyiadik aposztrófot, amennyi a kifejezés értékének egész része (ha nincs ennyiedik aposztróf, akkor a következő sorra megy), s ettől kezdve hajtja végre az utasításokat, amíg egy következő aposztrófot vagy sorvéget nem talál. Erre még bárki azt mondhatná, hogy jó, jó, de akkor itt a hagyományos ON utasítást:

ON kifejezés 'GOTO 1. sorszám 'GOTO 2. sorszám 'GOTO 3. sorszám '...

alakban kell írni? A válasz, hogy NEM! Ugyanis, ha az aposztrofok között mindenhol ugyanaz az értékadástól eltérő utasítás szerepelne, akkor ezt elég egyszer leírni az első aposztrof elé, tehát példánkban így:

ON kifejezés GOTO '1. sorszám '2. sorszám '3. sorszám'...

Lássunk most néhány példát az ON utasítás használatára:

1. A programban van 2 kapcsoló K és L, melyek értéke 0 vagy 1. Egy megfelelő helyen a négyféle kapcsolóállástól (K = 0, L = 0; K = 0, L = 1; K = 1, L = 0; K = 1, L = 1) függően négy különböző dolgot akartunk csinálni. Ezt a következőképpen tehetjük meg (persze az aposztrófok közé azokat az utasításokat kell írni, amiket az egyes esetekben végre akarunk hajtani).

```
100 ON 2*(K+1)-L'A=1: PRINT "L VAN BEKAPCSOLVA,";'A=0: PRINT "MINDKET K
APCSOLO KI VAN KAPCSOLVA,";'A=11: PRINT "MINDKET KAPCSOLO BE VAN KAPCSO
LVA,";'A=10: PRINT "K VAN BEKAPCSOLVA,";
110 PRINT " IGY A ERTEKE: ";A
```

2. Egy játékprogram elejét sokszor így lehet szépen megírni:

```
20 INPUT "A VERSENYZOK SZAMA  (1-6)? ";N
30 FOR I=1 TO N
40 ON I PRINT '"AZ ELSO";'"A MASODIK";'"A HARMADIK";'"A NEGYEDIK";'"AZ
OTODIK";'"A HATODIK";
50 INPUT " VERSENYZO NEVE: ";M$(I)
60 NEXT I
```

3. Nagyon jól ki lehet használni azt is, hogy a logikai műveletek (pl. az összehasonlítások, AND, OR) értéke 1 (igaz), vagy 0 (hamis). Tegyük fel, hogy egy játék végén P őrzi azt, hogy a játékosnak hányadik próbálkozásra sikerült valamit kitalálni. P értékétől függően ~~meg akarjuk~~ dicsérni vagy el akarjuk marasztalni. Ezt így tehetjük meg:

```
500 ON (P<11)+(P<13)+(P<17)+1 PRINT '"ABSZOLUT NEM MEGY NEKED EZ A JATE
K, INKABB PROBALJ KI EGY MASIKAT!"'"NEM REMEKELTEL TULSAGOSAN!"'"EGESZ
JO, DE MEG LEHET JAVITANI!"'"GRATULALOK, KIVALO A TELJESITMENYED!"
```

4. Szeretnénk készíteni egy rajzolóprogramot, aminek a következő dolgokat kell tudnia:

Először a képernyő közepén megjelenik egy pont. A pont az I, J, K, M billentyűk hatására mozogjon felfelé, balra, jobbra, ill. lefelé, és húzzon maga után vonalat. Persze vigyázni kell arra, hogy ne menjen ki a képernyőről, hiszen akkor hibajelzést kapnánk. Egy ilyen program a következő (a 10-es sor a képernyő grafikusra váltását és törlését, a 20-as sor a billentyűzet figyelését végzi):

```
10 CR=0:GL=200: PRINT CHR$(5):CR=1
15 X=160:Y=100
18 PLOT X,Y
20 A=USR(A): IF A<73 OR A>77 THEN GOTO 20
30 ON A-72'Y=Y-(Y>0)'X=X-(X>0)'X=X+(X<319)' GOTO 20'Y=Y+(Y<199)
40 GOTO 18
```

5. Végül még egy példa konkrét utasítások nélkül. Van egy 10 utasításból álló sorozatunk, melyet 6-szor kell egymás után végrehajtani úgy, hogy az 1–3. és 8–10. utasítások mind a 6 esetben megegyeznek, de az 5., 6. és 7. utasítások eltérőek Ezt a következő szerkesztőprogrammal lehet megvalósítani:

```
10 FOR I=1 TO 6
20 utasitas-1
30 utasitas-2
40 utasitas-3
50 utasitas-4
60 ON I'ut.5/1:ut.6/1:ut.7/1'ut.5/2:ut.6/2:ut.7/2'ut.5/3:ut.6/3:ut.7/3'
ut.5/4:ut.6/4:ut.7/4'ut.5/5:ut6/5:ut.7/5'ut.5/6:ut.6/6:ut.7/6
70 utasitas-8
80 utasitas-9
90 utasitas-10
100 NEXT I
```

Aki ezek után a példák után még kételkedik az AIRCOMP-16 BASIC ON utasításának hatalmas előnyeiben, az próbálja megírni valamelyik másik BASIC-ben a 2., 3. és 4. programot.

Király Zoltán

ROMOM K 6311 MÁTRIXNYOMTATÓ



SZÉP KIVITEL
ALACSONY ZAJSZINT
TISZTA NYOMTATÁSI KÉP!

ILLESZTÉSI LEHETŐSÉG COMMODORE 64 ÉS HT 1080Z SZEMÉLYI SZÁMÍTÓGÉPHEZ!

ALKALMAZÁSI TERÜLETEK:

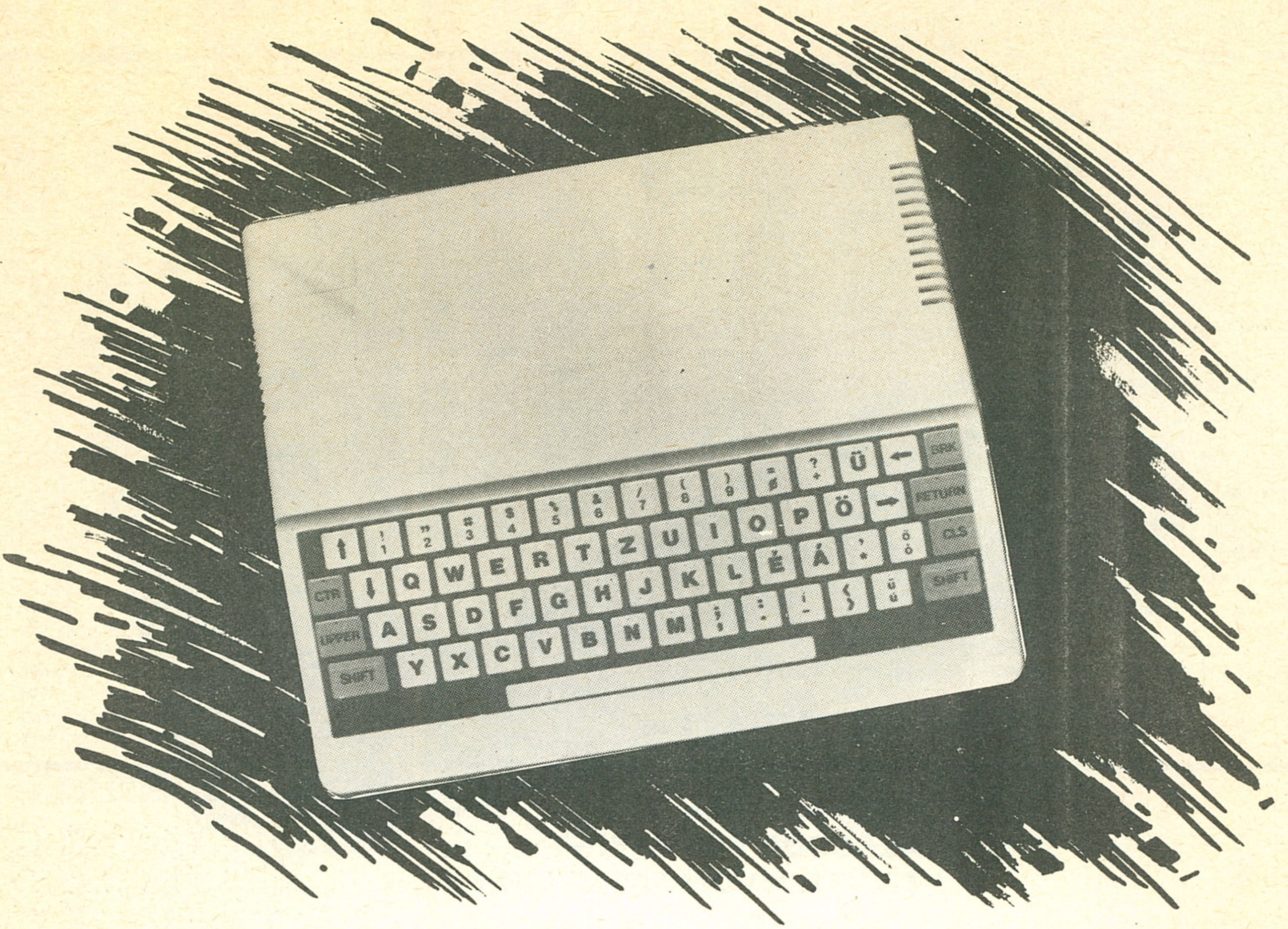
- PROFESSZIONÁLIS NYOMTATÁS
- KISSZÁMÍTÓGÉPES RENDSZEREK
- MÉRÉSADATGYŰJTŐ RENDSZEREK
- TERMINÁLOK KIMENETI NYOMTATÓJA.

MIGÉRT

GYÁRTJA:
MAGYAR OPTIKAI MŰVEK BP. XII., CSÖRSZ U. 35.

FORGALMAZZA:
MIGÉRT SZÁMÍTÁSTECHNIKAI OSZTÁLY
BP. VI., NÉPKÖZTÁRSASÁG U. 2. II. EM.
TEL.: 323-332

MEGTEKINTHETŐ ÉS MEGVÁSÁROLHATÓ:
MIGÉRT 4. SZAKÜZLET
Bp. VIII., RÁKÓCZI ÚT 57/A.
TEL.: 143-471



Jött... jött: a Primo. Az 1984-es év tavaszán tartott sajtótájékoztatón beígérték – nem igazán hittünk benne. A „jólértesültek" már szeptemberben jól értesülve ingatták a fejüket – hát ebből sem lett semmi, pedig közben a gyártó cég, a MICROKEY KFT csendben, majdnem titokban elkezdte az előrendelések kielégítését. És meglepő gyorsasággal három hónap alatt leszállítottak közel 2000 darabot.

GYÁRI ADATOK

Három kivitelben, különböző memóriamérettel gyártják a gépet:
32 kbyte 16 100 Ft
48 kbyte 20 500 Ft
64 kbyte 23 900 Ft
Méret: 45x260x310 mm
Súly: 800 gramm (tápegység nélkül)
Az alapgéphez normál kazettás magnetofon és fekete-fehér televízió csatlakoztatható, egyéb perifériák még nincsenek.

KÍNRENDSZER

A vallatáson a gyártó cég képviselői is részt vettek. Természetesen nem pontoztak, viszont néhány információval a vallatás szempontjait befolyásolták. Így végül is inkvizítoraink úgy döntöttek, hogy a szokásos kínrendszert alkalmazzák, egyetlen kivétel, hogy a perifériákat nem pontozzuk, mivelhogy nincs mit pontozni. Kiindulásnak az inkvizítorok is elfogadták a gyártók által megfogalmazott célokat: azt, hogy a konstrukció olcsó legyen, szocialista gyártmányú alkatrészekből épüljön fel, és tartalmazza a jövőbeli bővítés lehetőségét.

Ár – 4,1

1 KÍN

Nagy öröm, hogy egy magyar gyártmányú gép a hazai meglehetősen kusza árviszonyok között ilyen magas osztályzatot ért el. Általános vélemény az, hogy elfogadható az ár, hát persze jobb lenne, ha még olcsóbb lenne, de úgy tűnik, hogy hazai alkatrész-, munkaerő- és piaci viszonyok között ez már irreális. A gép ezen az áron megvásárolható, cégeknek, iskoláknak és megszállottaknak megéri – de még mindig nem az a gép, ami a háztartásokban tömegesen elterjedhet. Az azonban mindenképpen dicséretes, hogy megszületett az első hazai számítógép, ami a polcról vásárolható, tehát nem kell előre fél évre megrendelni, hanem lényegében készpénzért azonnal kapható. Inkvizítoraink megjegyezték, hogy a gép árához képest a tápegység egy kissé drága – 4600 forint.

Képernyőkezelés – 4,3

2 KÍN

Inkvizítoraink megegyeztek abban, hogy a képernyőkezelés egyszerű, jó minőségű – tudja azt, amit egy ilyen kategóriájú gépnek tudnia kell. Sőt! Néhány olyan dolgot is, amit kevés más berendezés tud. Feltétlen öröm, hogy a gép karakterkészlete a magyar abc majd valamennyi betűjét tartalmazza, amit ma, 1985-ben még a Képújság sem tud. Kétségtelen, hogy egyszerűbb mentegetőzni, mint új karaktereket definiálni. Még két érdekesség, amit külföldi gyártmányú személyi számítógépek sem tudnak általában: a függőleges írás – ami koordinátarendszerek feliratánál igen hasznos lehet, valamint az alsó indexbe írható bármilyen karakter, ami kémiai képletek írásánál szinte elengedhetetlen. Szőrösebb szívű inkvizítoraink megjegyezték, hogy ez mind igaz, de azért ma már a világszínvonalat a színes képernyő jelenti! És van egy-két olyan funkció, amit már kitaláltak, igaz, hogy a Commodore sem tudja, de azért a Primónak illene (pl. DRAW, FLASH)! Hát, igen...

Hang – 3,6

3 KÍN

A gépben levő egyetlen hanggenerátor, van akinek megfelel és elegendő, mások viszont kevésnek tartják. A programozás is ellentmondásos, egyes inkvizítorainknak egyszerű, mások bonyolultnak tartják. Feltétlen hiányosság viszont, hogy a hangerőt nem lehet szabályozni. A konstruktőrök elárulták, hogy a hangot eredetileg ki akarták hagyni a gépből, elsősorban azért került bele, hogy a billentyűzetet támogassa, jelezze a karakterek bevételét. A magyarázat elfogadható, bár ettől nincs jobb hangja a gépnek.

Kazettás tárolás – 3,9

4 KÍN

Az osztályzat nem rossz, különösen, ha azt is figyelembe vesszük, hogy hányféle, mennyire különböző kazettás magnetofonnal használják a

gépeket. Általában inkvizítoraink úgy tapasztalták, hogy kényes a magnetofonra, de ha egyszer egy gépet egy magnóval összeszoktattak, akkor utána már igen ritka a hiba. Előnyös és jól használható funkció, hogy egyetlen parancsra kimenti magnetofonra a képernyő tartalmát, és hasznos, hogy beolvasás közben hibajelzést is ad. Ettől ugyan nem jobb a beolvasás, de legalább lehet tudni, hogy hol van a hiba. Már a bővítésekre gondolva a gép bizonyos file-kezelési funkciókat is képes ellátni.

Gépi kódú programozás – 3,2

5 KÍN

Nem valami fényes osztályzat, de meg is van az oka. A gépben nincs monitor, a programozás pedig egy kissé nehézkes. Egyelőre hiányzik a megfelelő dokumentáció is ahhoz, hogy kényelmesen lehessen dolgozni vele.

Megbízhatóság – 3,9

6 KÍN

Általában megbízhatónak tartották inkvizítoraink a gépet, kisebb hibáktól eltekintve. Amit tapasztaltak, az nem általános hiba, hanem mindenütt más: az általános iskolai nyaggatás persze jobban megviseli a berendezéseket, mint egy otthoni, kíméletes kezelés. Egyik inkvizítorunk – talán kissé túlzóan úgy fogalmazott, hogy nincs is benne semmi, ami elromolhat! A gyártók ezt úgy módosították, hogy iparunk mai viszonyai mellett nincs is olyan, ami nem romolhat el! Dicséretes viszont az, hogy elsősorban információszerzés miatt a gyártó cég a kezdeti időszakban nem javítja a készülékeket, hanem minden elromlottat kicserél. Ehhez azonban az a kényelmetlenség is hozzátartozik, hogy – esetleg csak a tápegység hibája miatt – az egész gépet kell visszavinni, csomagolással együtt.

Billentyűzet – 3,7

7 KÍN

Minden olyan kisszámítógépnél, ahol nem professzionális típusú billentyűzetet használnak fel (pl. Commodore) ez az egyik legfontosabb kérdés. A konstruktőrök szeretnének valamilyen olcsó megoldást találni, ezért igen sokat bűvészkednek, végül megszületik a klaviatúra, ami ugyan olcsó, talán még jó is, de senkinek sem tetszik, mert nem olyan, mint a profi gépeké. A Primo – nyilvánvalóan anyagi okokból – nem profi billentyűzettel készül. Ellentétben azonban a Sinclair gépekkel, nem is fóliaérintkezős, amely mozgó alkatrészeket tartalmaz, tehát előbb-utóbb biztosan elromlik. A Primo kapacitív érzékelő billentyűkkel rendelkezik, ami sokak szerint a legjobb, strapabíró, klímaálló, jó konstrukció. A hibák többsége a helytelenül beállított érzékenységből adódik. A gyártók mérései szerint a beállítás két véghelyzete a következő: érzéketlen beállításnál egy elefánt sem tud rajta dolgozni, ellenkező esetben viszont kézrátétel nélkül, egyszerű „szemmelveréssel" is beindul a gép, magától. A két véglet között viszont megtalálható lenne a megfelelő, amit a gyárban be is tudnának állítani. A kérdésre, hogy akkor miért nem állítják be, rövid volt a válasz:

– Azért, mert az üzemben nem japánok dolgoznak!

Mindezek ellenére az általános vélemény az volt, hogy bár kezdetben szinte mindenki idegenkedett tőle, többen megkedvelték, különösen mert megfelelően nagy méretűek a billentyűk.

Dokumentáció – 2,9

8 KÍN

Valaki azt írta, nem is érdemes beszélni róla, és az osztályzat egyértelműen tükrözi: lehet, hogy gépet már tudunk gyártani, de dokumentációt még biztosan nem. Sok minden nincs benne, ezek egy részét szándékosan nem publikálták a készítők, más része egyszerűen kimaradt. Hiányzik a védett kulcsszavak jegyzéke, a BASIC utasításkészlet jegyzéke, ráadásul, ami benne van, azt is többen írták, így stílusában is „változatos". A géphez kapott demonstrációs kazetta értelmes dolog, de jó gépkönyv nélkül ez is kevés. A géphez tartozó dokumentáció helyzetén ugyan nem javít, de feltétlenül hasznos lesz, hogy beindulnak a Primo-füzetek, amelyben a gyártók valamennyi titkukat fel fogják tárni, sőt, mások ötleteit is várják és közzéteszik.

Editálás – 4,0

9 KÍN

Erősen megoszlottak a vélemények ebben a témában. A gép nem rendelkezik teljes képernyőn való javítási lehetőséggel (fullscreen editor). Így néhányan eleve gyengének tartják, akik ehhez szoktak hozzá. Mások viszont a meglevő soronkénti javítási lehetőséget egyszerűnek, könnyen kezelhetőnek tartják. Beíráskor a gép nem végez szintaktikai elemzést, így a hibák futtatás során jönnek elő, akkor viszont – hiba esetén – a gép automatikusan editor üzemmódba kapcsol, kiírva a hibás sort. Egyöntetű vélemény volt, hogy hiányzik a RENUMBER és a MERGE.

A gép programnyelve – 4,1

10 KÍN

Van, aki a világ legjobbjának tartja! Van, aki gyengébbnek, de abban szinte mindenki egyetértett, hogy jól szervezett, jól felépített nyelv, amelyből szinte semmi sem hiányzik, ami nélkülözhetetlen lenne. Számunkra is őszinte öröm, hogy magyar tervezésű BASIC-ről ilyet írhatunk.

Tanulhatóság – 4,3

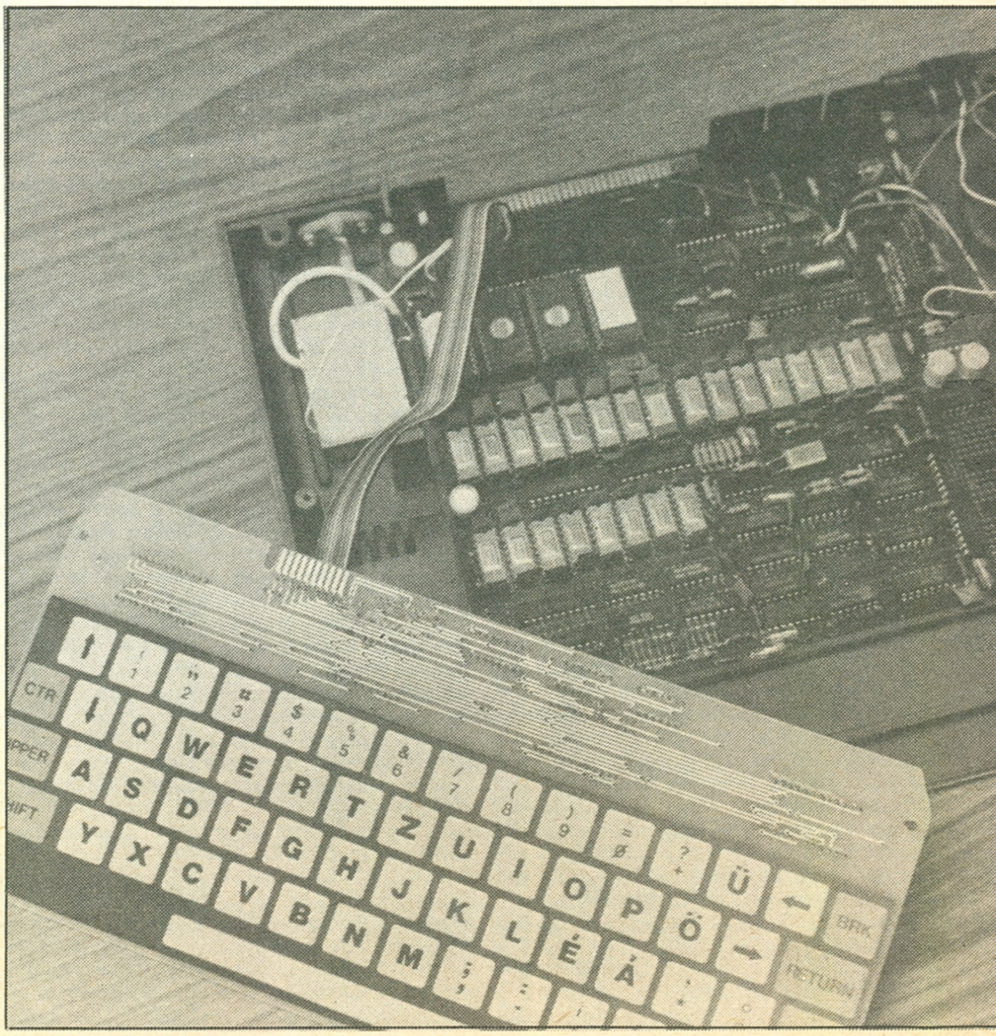
11 KÍN

Igen jó osztályzat, ami azt jelenti, hogy a gép könnyen elsajátítható, egyszerűen kezelhető. Valaki úgy fogalmazott, hogy néhány óra alatt még a szakember is megtanulja!

Emberközelség – 4,0

12 KÍN

Bár az osztályzat nem rossz, úgy érezzük, hogy ennél jobb is lehetne. Ebben a kínban általában azt igyekszünk osztályozni, hogy a gép menynyire kínálja a lehetőségeket, a kényelmes megvalósítási módszereket. A Primo tartalmaz ilyen lehetőségeket, azonban ezek nagy része még nincs publikálva, a felhasználónak kell rájönni. Vallatásunkon is érte meglepetés a konstruktőröket: egyik inkvizítorunk bemutatta, hogy a gép rövidített módon is elfogadja a Basic kulcsszavakat, csak tudni kell, hogy mit gépeljenek be. A konstruktőrök nem tudták.



PRIMO VALLATASANAK EREDMENYE 1985. JANUAR 5.	CZIFRA ZOLTAN GIMNAZIUMI TANULO	DR. SZEKELY JENO FOISK. TANAR	GELLERI PETER ALT. ISK. TANULO	GARADI JANOS MATEMATIKUS	BOGLAR GYULA QUARZ GMK. TAG	KHIN LASZLO FOISK. HALLGATO	BRANYI LASZLO FOISK. HALLGATO	AMTMANN TAMAS SZERVEZO	AMTMANN ARPAD VILL. MERNOK	
K I N O K	1	2	3	4	5	6	7	8	9	ATLAG
1 .KIN:AR	4	4	3/4	4/5	5	4/5	3/4	4	4	4.1
2 .KIN:KEPERNYOKEZELES	4	4/5	5	5	4/5	4	3/4	4	4	4.3
3 .KIN:HANG	3	-	3	4	2/3	4	3	5	4	3.6
4 .KIN:KAZETTAS TAROLAS	3/4	-	4	4	5	3/4	3	4/5	4	3.9
5 .KIN:GEPI KODU PROGRAMOZAS	3	4	3	-	3/4	-	2/3	-	3	3.2
6 .KIN:MEGBIZHATOSAG	4	4	4	3/4	5	3/4	3/4	4	4	3.9
7 .KIN:BILLENTYUZET	3	4	2/3	4	5	3/4	3	4	4	3.7
8 .KIN:DOKUMENTACIO	3/4	4	3	3/4	3/4	3/4	2/3	1	2	2.9
9 .KIN:EDITALAS	5	4	5	4/5	4/5	4	3	3	3	4.0
10 .KIN:A GEP PROGRAMNYELVE	4	4/5	4/5	5	5	4/5	2	3/4	4	4.1
11 .KIN:TANULHATOSAG	4	5	4	5	5	5	3	4	4	4.3
12 .KIN:EMBERKOZELSEG	3/4	4/5	4	4/5	4/5	4/5	3/4	4	3	4.0
+ 1 KIN:SZUBJEKTIV VELEMENY	4	4	4	4/5	5	4	3	4	3/4	4.0
ATLAG	3.7	4.2	3.8	4.3	4.5	4.0	3.0	3.7	3.6	3.9

Szubjektív vélemény – 4,0

+1 KÍN

Az osztályzat sok magyarázatot nem kíván, inkább csak azt érdemes megemlíteni, hogy a négyesre az ember hajlamos azt gondolni, hogy nem ötös, tehát rossz. Néha eszünkbe kell hogy jusson az is, hogy nem hármas.
Eddig tartott a Vallatás, ami ezután következik, az inkább a szerénytelen jóslat műfajába tartozik. Szerkesztőségünk nem kíván reklámot csinálni, netán döntőbíróként szerepelni a Spectrum- és a Commodore-hívők vitájában. Sőt! Most azonban mégis úgy érezzük, hogy a Primo megjelenése többet jelent annál, hogy még egy számítógéptípus kapható az országban. Többet jelent – ha nem lenne annyira semmitmondó a kifejezés, akkor úgy fogalmaznánk: mérföldkő. Így inkább azt mondjuk: valaminek a kezdete lehet, itthon elnyerheti azt a szerepet, amit a Commodore vagy a Spectrum kapott a fejlett országokban. Megvan a lehetősége arra, hogy tömegesen elterjedve egy személyiszámítógép-kultúra alapjait jelentse, hogy már ne azon vitatkozzunk, hogy honnan és hogyan szerzünk gépet, hanem azon, hogy mire is lehet használni! És ez mindenképpen nagy lépés.

HOZZÁSZÓLÁS

Felfokozott várakozással vettem a kezembe az Ötlet 4. évf. 5. számát, és mindjárt a BIT-LET-hez lapoztam ... És valóban ... végre ... Primó- vallató. Szinte faltam a betűket. A vallató végére érve kihúztam magam, és mellem dagadt a büszkeségtől. Igen, eddig csak sejthettem, de most már biztos lehetek benne, hogy az, „aki" itt fekszik előttem az asztalon – a PRIMI(TÍV), merthogy így hívom – a világon jelenleg forgalomban levő gépek élcsoportjába tartozik! Az éles szemű szerkesztő ezek után rögtön konstatálhatja, hogy egy PRIMO-tulajdonos esete forog fenn. Nincs mit tagadni, bevallom, valóban a birtokomban van egy PRIMO A-32-es „PERSONAL COMPUTER". Mivel 1984 novemberében vásároltam a gépet, állíthatom, hogy az elsők között voltam, aki megelőlegezte a bizalmat egy teljesen ismeretlen, külsőleg abszolút visszataszító, korszerűtlen (fekete-fehér) – halmazatilag – magyar gyártmányú gépnek.
Ami igaz, az igaz, nem én jegyeztem elő a gépet, de ismerősöm, aki előjegyeztette, elállt a vásárlástól. Tudj' isten, talán időközben meglátott egy számítógépet, amit nem itthon készítettek. Én ennek ellenére úgy döntöttem, lesz ami lesz, ennyi pénzért megveszem a PRIMO-t.
Nosza menjünk, mondtam magamnak és az öcsémnek (ő egyszer már próbált is egy gépet a BNV-n), no meg a közben összejött 11 500 Ft-nak. Vittünk magunkkal egy papírt is, amelyen ismerősöm lemondott a vásárlási jogáról és a tápegységről.
Megérkeztünk Elektromodulékhoz a Jászai Mari térre. Egy hölgy fogadott. Udvariasan és fáradhatatlanul válaszolt – utólag már be kell látnom – együgyű kérdéseinkre. Gondolom, hogy lesírt rólunk a hozzá nem értés, ezért a hölgy adott néhány – a Primó-t reklámozó – szórólapot, és azt mondta, ha kedvünk van, próbáljunk meg összebarátkozni a helyiség egyik asztalára kitett géppel. Kaptunk egy tájékoztatót is, amiből már azt is megtudhattuk, hogy tápegység nélküli vásárlás esetén a gépre nincs garancia. Úgy voltunk vele, hogy ráhúzunk a tizenkilencre, és megvesszük a gépet garancia és tápegység nélkül, mert úgy gondoltuk, hogy otthon egy óra megfeszített munkával összeütünk egy olyan tápegységet, mint a PSA 01.
Utunk ismét a kedves és szolgálatkész hölgyhöz vezetett, és közöltük vásárlási szándékunkat. Közben a pultról az öcsém kezébe került a PRIMO felhasználói viccgyűjteménye. Míg én a vásárlás adminisztratív részét intéztem, öcsém a már említett viccgyűjteményt böngészte, amit a gyártó számunkra ismeretlen okból „FELHASZNÁLÓI KÉZIKÖNYV"-nek gúnyol. Egyszer csak felkiáltott: „ÁLLJ!" Mi történt, kérdeztük a hölgyeljel szinte egyszerre. Az a gyanúm – így az öcsém –, hogy a gép képmodulátora a tápegységbe van szerelve, és hiába is táplálnánk a gépet a legszebb egyenárammal, ami csak szűrésileg és stabilizációilag elképzelhető, akkor sem fog működni, mert nem tudjuk a tv-re kötni. Onnan, ahol én álltam, szintén hasonlóan nézett ki a dolog. Sokáig nem is vergődtünk kétségek között, mert a hölgy biztosított róla, hogy öcsém gyanúját bátran tényként kezelhetjük. Nem részletezem tovább, inkább röviditek: Hölgy: papírokat megírt; mi: gépet megtekint (kiválaszt); hölgy: számlát kézbe nyom, mi: 16 100 Ft-tal könnyebbek, gép hón alá, köszönés, hölgy: visszaköszön!!
Otthon: én kávét főzök, az öcsém összeállítja a „konfigurációt". Az öcsém felkiált. Én rosszat sejtek. Nemhiába. Gyártókámék újabb meglepetése: a modulátor nincs a tápegységben. A gépben van!

BOGLÁR GYULA: „Ebben a kategóriában ez a legjobb BASIC!"

GELLÉRI PÉTER: „Elég megbízható, de néha a hajamat téptem, mert a gép egyszerűen nem hozott be programokat még a DEMO kazettáról sem!"



Két nap múlva – egy program beírása közben – tápegység el. Villany be, de ki egy darab se. Reszkető kézzel töröm fel a sarkon levő festékzárat. Garancia el. Meglepve tapasztaljuk, hogy kitokozás után, egy fikarcnyi nem sok, de annyi aranyrúd sem hullik ki belőle. Őszintén szólva az ára után legalább egyre számítottunk. Ilyen kevés semmit ekkora dobozba rakni! A konstruktőrök újabb furfangja: primer oldali biztosíték házilagos cseréje csak a garancia elvesztésének testén keresztül. Szellemes. Biztosítékcsere után azóta is kifogástalanul működik. Mellékszolgáltatásként vizet melegít, tojást süt és fűti a lakást. Hangsúlyozni kívánom, hogy nem túlbiztosítás miatt. A kicserélt biztosíték értéke azonos az eredetivel. Itt visszakanyarodok a BIT-LET-beli PRIMO-vallatóhoz, hiszen az előzőeket csak bevezetőnek szántam. Talán az eddigiekből is kiderült, hogy nem mindenben csatlakozom a PRIMÓ-t vallató inkvizítorokhoz.

1. kín: ár. *A gép árát megfelelőnek tartom, de csak a 16 K-ét. A 32, 48 K-s gépek ára – nekem úgy tűnik – nem áll arányban a többletmunka és többletalkatrész árával. Most jön eddigi vesszőparipám, a tápegység. Inkvizítoraik (szemérmesen) úgy fogalmaztak, hogy „a gép árához képest a tápegység egy kissé drága". Én úgy gondolom, hogy nemcsakhogy kissé, de még csak nem is szemérmetlenül, hanem felháborítóan drága! Gondoljuk csak el, egy Junoszty televízió ára 3370 Ft. Egy BRG magnó 2900 Ft. A táp 4600 Ft-os ára a szabadrablás kategóriájába tartozik, és népi ellenőr után kiált! Többek szerint ha ennek az árnak a negyedrészéből egy kezdő amatőr nem épít legalább kettő jobbat, dobja el a pákáját!*

Végül van még egy problémám. Amikor benne volt a modulátor, 4600 Ft volt az ára. Az enyémben már nincs, mégis ennyi... Lehet, hogy ez korábban a gyártó ajándéka volt, mert akkor én kérek még egyet.

2. kín: képernyőkezelés. *A szőrösebb szívű inkvizítoraiknak üzenem, hogy ennyi pénzért örüljenek, hogy a fekete mellett még a fehéret is tudja. Bár a világszínvonal és a jövő valóban a színes. Egyébként teljesen egyetértek velük.*

3. kín: hang. *Az nulla. Ennél még az is jobb lett volna, ha a konstruktőrök ragaszkodnak az eredeti elképzelésükhöz és kihagyják, mert akkor legalább nem volna az embernek az az érzése, hogy van. Szerintem egy ilyen színvonalú gépnél első a grafika és rögtön utána a hang. Gondolom a tervezők sem gondolták komolyan, hogy ezt a gépet tudós emberek tudományos munkára fogják használni. Ez a gép jelenleg ilyen célra a jó perifériák és megfelelő tasztatúra hiányában alkalmatlan. Jelenleg a gép leginkább hobbycélokra, főleg játékra alkalmas, ahol egy jó hanggenerátor a gép egyik legfontosabb eleme.*

4. kín: kazettás tárolás. *Én, kérem, próbáltam „szoktatni" mindenféle magnóhoz (9-féle van itthon). A fene a gusztusát, csak a 14 000 Ft-os JVC kell neki. Az viszont igaz, hogy ezzel a hifi sztereó deck-kel „összeszoktak". Viszont az is igaz, hogy ha ez nem lenne, akkor bajban lennék, mert kisebb kaliberű magnókról 10-ből 8-szor hibásan jön be a program még a DEMO kazettáról is! Apropó, DEMO kazetta! A kazettának csak az egyik oldaláról lehet hibátlanul betölteni, s hogy melyikről, azt csak a gyakorlatból lehet megtudni, mert erre vonatkozó utasítás sehol sincs. A jó oldala is olyan, hogy amikor bekapcsoltam a JVC-t, a kivezérlést jelző műszerek „kifeküdtek" a +6 dB-nél, és szerintem még tudnának menni egy kört, ha a határoló engedné. Ahogy elnéztem a szalagot, egyszerű LH szalag van a kazettába töltve. Ilyen kivezérlés láttán csoda, hogy a PRIMO megbirkózik a torz jelekkel.*

5. kín: gépi kódú programozás. *Ehhez semmit nem tudok hozzászólni, mert örülök, ha BASIC-ben megértjük egymást a PRIMI-vel. Egy hozzáértő viszont azt mondta, hogy memóriatérkép nélkül félkarú óriás a gép.*

6. kín: megbízhatóság. *Erre csak azt tudom mondani, hogy ha azt kibírta, amit az első napokban éjt nappallá téve elkövettünk rajta, akkor olyan, mint a selyemhernyó – bírja a gyűrődést.*

7. kín: „billentyűzet". *Tartok tőle, hogy nem is beszélhetünk billentyűzetről, hiszen billentyűnek nyoma sincs a gépen, kivéve az egy RESET gombot, ami szintén roppant szellemesen a gép hátuljára került. (Csodálom, hogy a gép alja elkerülte a tervezők figyelmét.) Beszélhetünk azonban tappantyúzatról, masszírozantyúzatról és idegbajról, amit kapni lehet tőle! Én igazán irigylem azokat az inkvizítorokat, akik ezt a masszírozmányt „megkedvelték". Nekem ez három hónap alatt még nem sikerült. Attól tartok, hogy ezek az inkvizítorok rövid idő alatt megkedvelnék a keserű cukrot, a fokhagymás nápolyit és talán a mákos pacalt is. Mondom, nekem nem megy a megkedvelés, pedig egy egyszerű megoldással elértem, hogy az eredeti érzékenység kb. tizszeresére nőtt, annak veszélye nélkül, hogy a gép „szemmelveréssel" beinduljon. Azt is megfigyeltem – bár lehetséges, hogy megfigyelésem minden tudományos alapot nélkülöz –, hogy a három gyermekem közül a legkisebb (8 éves) tudja a legkevésbé aktivizálni a tappantyúkat. Ezt arra vezetem vissza, hogy ő rendelkezik a legkisebb testfelülettel (kapacitással??). Erről viszont az jut eszembe, ha a fenti megfigyelésem és következtetésem igaz, valamint igaz az is, hogy ezt a gépet akarják elterjeszteni az általános iskolákban, bizony könnyen meglehet, hogy egy teljes generációval megutáltatják egy életre a számítógépet. Tudniillik nincs annál lehangolóbb, mint amikor egy ügyességi játék során a gyerek biztos benne, hogy a megfelelő gombot a megfelelő időben nyomta le és a gép mégsem reagál. Mit mondjak neki ilyenkor azok után, hogy*



fél órával azelőtt még arról beszéltem neki, hogy a számítógép azért nagyszerű, mert szinte mindent meg lehet csinálni vele, és mivel csak a tudást nézi, roppant igazságos is. Mondjam talán azt, hogy meg kell várnunk, amíg japán vendégmunkások érkeznek hazánkba? Tartok tőle, hogy az az ember, aki ezt a japánokkal kapcsolatos kijelentést tette, igazán zaklatott lenne, ha mondjuk én 1 szárú nadrágot varrnék neki, pusztán azért, mert én sem vagyok japán.

8. kín: dokumentáció. *Ha teszem azt, vennék egy lábhajtású habverőt, az biztos, hogy komolyabb dokumentációt kapnék hozzá. Az előzőekben már egyszer említettem a Felhasználói kézikönyv címre hallgató viccgyűjteményt. Ezt azért neveztem el így, mert szerkesztői mindent elkövettek, hogy próbára tegyék rejtvényfejtői képességemet, és ahol csak lehet, megvicceljenek. Ráadásul minimum a matematikai tudományok doktorának kellene lennem, hogy megértsem. Néhány vicc (nem sajtóhiba!) a teljesség igénye nélkül: 2. és 4. ábra. Az ábrák alapján a rendszer nem állítható össze.*

16. old. „A 7 db zöld alapszínű billentyű..." Tűvé tettem a gépet zöld alapszínű „billentyű"-ért. Eredménytelenül.

17. old.: „A billentyűzet az érintésre kellően érzékeny." Ezt csak innen lehet megtudni, mert ugye a tapasztalat...

19. old.: „Figyelembe ajánljuk a 4. mellékletben található DEMO programlistát." Beírtam a Rajzoló-t. Nem működött. Hibás a program.

22. old.: „Mivel Ön már ismeri a PRIMO számítógép kezelését..." Persze, mert én egy zseni vagyok!

35. old.: „A listázás folyamatosan történik és a karakterrel szakítható meg." Esküszöm, hogy fordítva van!

60. old.: Hiányzik a megengedett vezérlő kódok közül a függőleges kiírásé, valamint ennek inverze.

70. old. Hiányzik a függőleges írás kódja (CHR$ (15) inverze CHR$ (23)).

A gyártóknak a „titkaikat" nem a gép megjelenése után fél évvel kell feltárni, hanem egy normális gépkönyvbe foglalni, és a gép mellé kell adni! Ráadásul úgy megfogalmazva és illusztrálva, hogy egy teljesen kezdő is megértse.

9. kín: EDITÁLÁS. *Egyszerű de nagyszerű!*

10. kín: programnyelv. *Én a magam részéről csodálatosnak tartom.*

11. kín: tanulhatóság. *Egyetértek az inkvizítorokkal.*

12. kín: emberközelség. *Úgy mint fentebb.*

+1 kín: szubjektív vélemény. *Ezt inkább kihagyom.*

Nos, végére értem a vallatásnak. Döntsék el Önök, hogy van-e igazságom, s ha van, mennyi. Bár én számokkal külön nem osztályoztam az egyes kínokat, mégis a végére ideírom azt, hogy összességében 2,5-re értékelem az egész ketyerét úgy ahogy van. Meggyőződésem, hogy az Önök 3,9-es osztályzata a túlzott hurrá optimizmusukból fakad. Az ilyen optimizmus nem sarkallja jobb munkára sem a tervezőket, sem a gyártókat. Sőt, éppen ellenkezőleg, a jól végzett munka utáni pihenés lehetőségét sugallja.

Végül egy utolsó megjegyzés. *Igazán nem rosszindulatból kérdezem. Nem véletlenül most törleszt a BIT-LET a PRIMO DEMO kazettáján levő reklámért?*

Én például azóta veszem a lapot, amióta azt láttam.

Szőke Ferenc
1108 Harmat u. 182. 1/2.

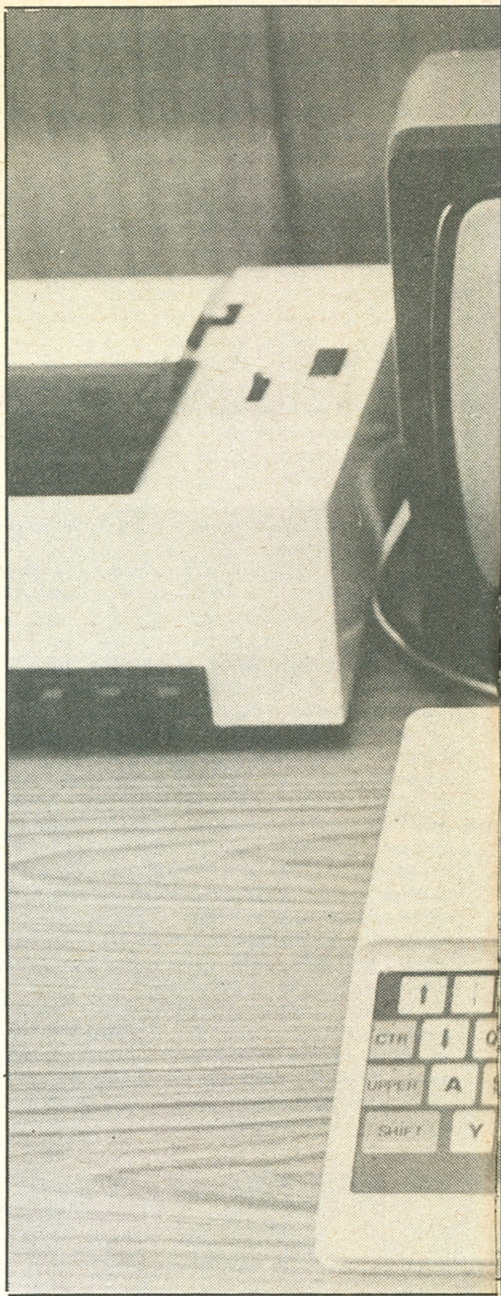
Válasz: nem. Már törlesztett – lásd PRIMO beharangozás 1984. május. Egyébként Önnek sok mindenben igaza lehet, s ráadásul milyen jól szórakoztunk élvezetes stílusán!

Már megint a magyar ABC!

Szaklapjainkban már sokszor írtam arról, hogy nincs olyan magyar gyártmányú, vagy itthon a kereskedelemben kapható mikroszámítógép, amelyik képernyőjén a teljes (és ezt a szót hangsúlyozom) magyar betűkészletet képes lenne kijelezni. (Természetesen itt nem arra gondolok, hogy a betűket mint rajzokat képes előállítani. Ezt minden rajzmegjelenítésre képes gép tudja.) Hiába! Most megjelent a BIT-LET 1985. január 11-i számának PRIMO-vallatójában az: „Feltétlen öröm, hogy a gép karakterkészlete a magyar abc összes betűjét tartalmazza." Ez a fogalmazás megtévesztő, ugyanis az összes kisbetűt tudja csak, de nem a hosszú ékezetes nagybetűket. Ez pedig nem „a magyar abc összes betűje".

Azt, hogy ebben nekem és nem a „Vallató" rovat írójának van igaza, azt a BIT-LET ugyanezen számának 30. oldalán levő cikk is alátámasztja azt írva: „képes a magyar nyelv valamennyi ékezetes kisbetűjének és az ékezetes nagybetűk egy részének megjelenítésére". Ez a cikk viszont azért igényel helyreigazítást, mert az nem felel meg a valóságnak, hogy „a PRIMO az első olyan hazai személyi számítógép, amely képes a magyar nyelv valamennyi kisbetűjének megjelenítésére". A SIMON 68 ezt már kb. egy évvel a PRIMO előtt tudta, és tudta – mégpedig helyes rendezésben – az összes magyar betűt!

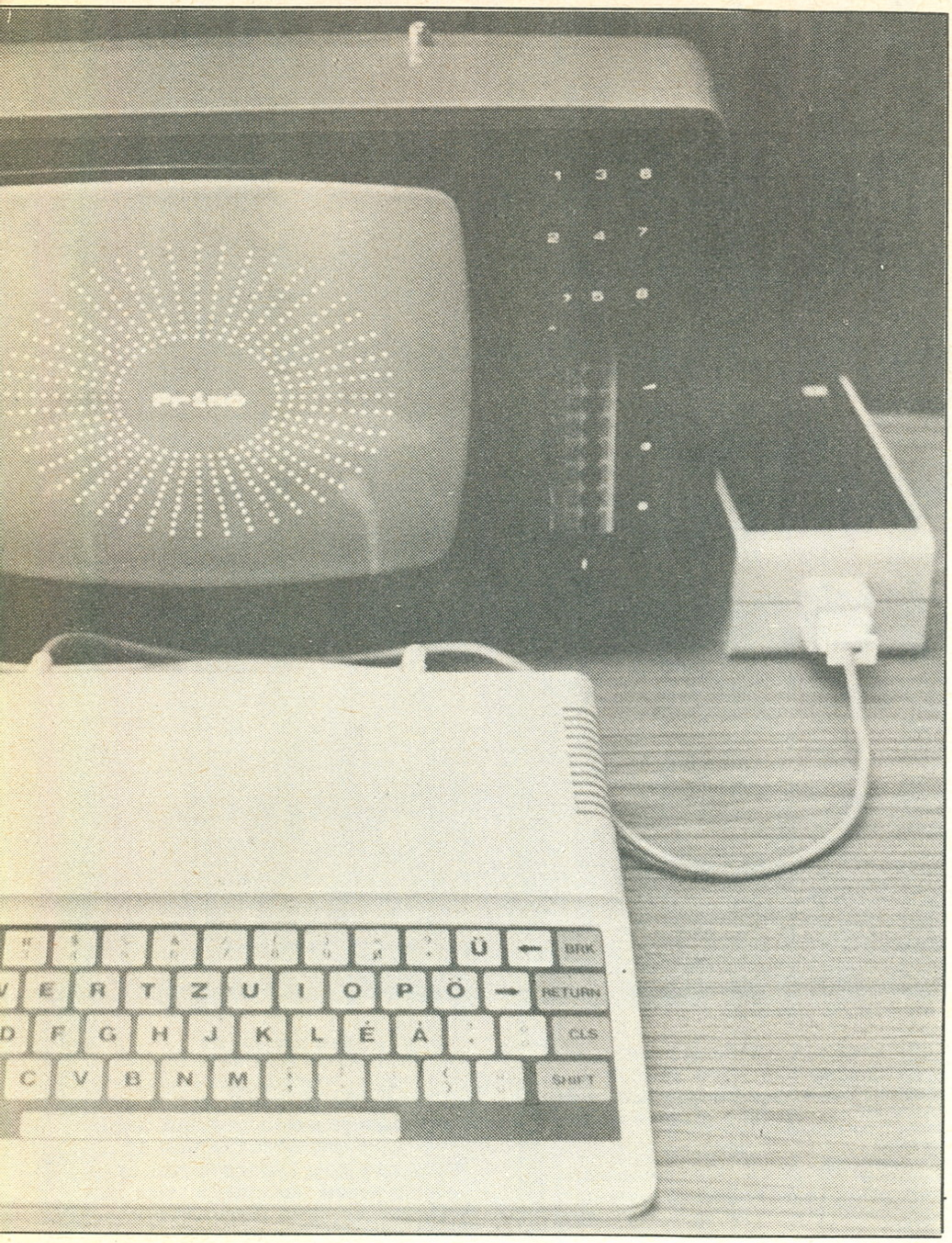
A magyarázat arra, hogy miért lehetett a SIMON '68-nál, vagy a DRAGON '64-nél megoldani ezt a problémát, míg a többinél nem, nem az, hogy az én képességeim felülmúlják az összes többi magyar szakemberét, hanem a hardverlehetőségek különbsége. Az általam említett gépekben a karaktereket az MC 6845, illetve a 6847 típusú karaktergenerátor, és egy 2716 típusú EPROM állítja elő. Ezen karaktergenerátorok viszont képesek 8x16 pontból álló, ún. betűmátrixot használni. Ezzel szemben a PRIMO 6x12, a Commodore és Sinclair gépek 8x8, a HT 1080 Z (teljes vagy félsor kijelzéstől függően) 4x12 vagy 8x12 pontos mátrixot használ – amiben, a feltétlenül kihagyni szükséges betű és sorköz miatt – nem fér el minden betű, és nem helyezhető el a magyar helyesírás szabályai szerint!



Miért foglalkoztat ennyire ez a probléma, hogy újra és újra írok? Azért, mert ilyen gépekkel nem lehet sem helyesírást tanítani, sem helyesen írni! Tessék tehát megtanítani a magyar gépeket magyarul! Megvalósítható ez a követelmény? Igen, mert mindkét fajta IC olcsó, nagy tömegben gyártott (a 6845-öt Bulgáriában is gyártják); nemcsak 68XX (vagy 65XX) típusú mikroprocesszoros számítógépekben, hanem Z80-asokban is használhatók, így a legtöbb hazai gép átalakítható lenne!

Végül, hogy ne csak bíráljak, feladatot adjak: a HCC klub vállalja – megkeresés esetén – bármely itthon nagy mennyiségben kapható mikroszámítógéphez a teljes magyar betűkészlet helyes elrendezésben kijelezni képes megoldás kidolgozását.

dr. Simonyi Endre
1125 Bp. Trencsényi u. 19.



Olvastam a PRIMO vallatását. Elsősorban ehhez szólnék hozzá. Összességében én a géppel meg vagyok elégedve (PRIMO A–64), azonban a hozzá kapott tápegységgel egyáltalán nem. Ahogy Amtmann Árpád írta: „Bírja a 72 órás kikapcsolás nélküli üzemet!” Hát ehhez én azt mondanám; a gép igen, de a tápegység egyáltalán nem. A tápegység egy hét (kíméletes) használat után bemondta az unalmast, ki kellett cserélni. A második egység eddig még ilyen szempontból jó, de a géphez jövő csatlakozója olyan rosszul van beforrasztva, hogy ahogy mozgatom a vezetéket, a kép borzasztó sokat változik, sőt néha tiszta fehér a kép, még reszetre sem áll vissza normál üzemmódba, s ha nem mozdítom meg ismét a vezetéket, akkor örökké csak a nagy fehér semmit látom. Ha szerencsém van, akkor a vezeték olyan állapotban van, hogy a kép nagyon-nagyon szép (3 hónap alatt kétszer).

***Editálás.** Négyes osztályzatot kapott a gép, holott szerintem a PRIMO-nak ez a leggyengébb pontja. Néha, ha megmakacsolja magát, nem engedi átjavítani a hibás sort. Illetve átengedi, de vissza is írja. Ekkor általában az sem segít, ha újrairom a sort. Ilyenkor a legbiztosabb módszer, ha megkeresem, hol van a tárban az a bizonyos hiba és POKE-kal javítom. Kissé hosszadalmas. A másik ok, amiért írok, a következő: a Vallatóban azt írták, hogy elkészült vagy előkészületben van a gépkönyv, szoftvertérkép, assembler, disassembler, Forth, Pascal. Én igényt tartanék ezekre. Ha kell, kazettát is küldök, s postán a programnak és a többinek az árát. Visszatérve magára a gépre, valóban elég jó teljesítményű kisgép, eltekintve ettől a pár hibától (EDIT). Valóban hiányzik belőle a DRAW és a FLASH utasítás, de ezeket kis ügyességgel a PRIMO-n könnyű pótolni, még BASIC-ben is. Valóban jó tulajdonsága a gépnek, hogy a kulcsszavakat rövidített formában (különböző CHR$-ben használatos jelekkel) is elfogadja. Csak ugye néha egyszerre három billentyűt úgy fogni, hogy a lehető legnagyobb felülettel érjünk hozzá (vagy négyet)! SHIFT+↓+egy billentyű, vagy SHIFT+↓+CTR+billentyű.*

egy PRIMO-tulajdonos: ***Fehér Csaba,***
8360 Keszthely, Fürst S. u. 10.

Nagy érdeklődéssel vettem kezembe a BIT-LET-et, ugyanis érdekelt, hogy mit írnak a PRIMO-ról. A cikk végigolvasása után azonban egy kicsit dühös voltam a cikk írójára. Ez nem csoda, hiszen olyanokat írt, amik joggal felháboríthatták bármelyik jobbérzésű PRIMO-tulajdonost. Engem konkrétan két dolog bántott.

Az egyik a kazettás tárolással kapcsolatban írottak. Nem tudom, hogy milyen „kaliberű” magnókkal rendelkezik a tisztelt író, de nekem egy MK 29-es magnó nagyon jól működik. Ez, azt hiszem, eléggé kis kaliberű. Sem a felvételnél, sem a lejátszásnál nem volt még probléma. Még a DEMO kazettánál sem. Egyébként a DEMO kazettán meg vannak jelölve az oldalak, és valaki könnyen megjegyezheti, hogy a B oldal a hibás oldal.

A másik dolog a billentyűzet. Abban egyetértek Szőke Ferenccel, hogy a PRIMO billentyűzete nem a legkedvezőbb, azonban megfelelő beállítás esetén nem okoz gondot a gépelés (nálam legalábbis). A játékprogramoknál tényleg gondot okoz a billentyűzet, azonban gondoljunk csak arra, hogy a számítógépet elsősorban nem játékra kell használni. A dokumentációval kapcsolatban nekem is az a véleményem, amit a cikkben olvashattunk. A jegyet azonban egy kicsit erősnek tartom. A gép szerintem bőven megüti a 3-as szintet, ha ennél nem is jobb sokkal. Egy megjegyzés. A PRIMO-hoz közölt (illetve nem közölt) programoknak a túl kevés számát jó lenne gyarapítani.

És végül, hogy ne csak kritizáljak – egy észrevételemet közkinccsé teszem.

Próbáljuk ki a következő programot:

```
10 CLS
20 FOR I=1 TO 2
30 POKE 16457, I
40 ?"PRIMO PERSONAL COMPUTER"
(Ide bármilyen szöveget lehet írni)
50 NEXT
```

Meglepve tapasztalhatjuk, hogy a képernyő aktuális helyére kiírt szöveg karakterei megvastagodnak. Észrevehetjük azonban azt, hogy a szöveget nem lehet PRINT $-ral kiíratni. Csak úgy, hogy a szöveg elé megfelelő számú PRINT-et írunk, majd ha elértük az aktuális sort, ott PRINT TAB(x) "szöveg"-gel kiíratjuk a képernyőre az üzenetet.

Próbáljuk ki a programot úgy, hogy a ciklusváltozó második értéke ne 3 legyen, hanem annál nagyobb. Valamint úgy, hogy a ciklus 180–210-ig terjedjen. Ezt a programot ki lehet egészíteni különböző vezérlő karakterekkel, az eredmény azonban nem minden esetben változik. Remélem, levelem nem talál süket fülekre, és nem kerül a szemeteskosár mélyére!

Nógrádi Előd
(2. oszt. gimn. tanuló, Szeged)

NOVOTRADE

RT

az alábbi könyveket, programokat és hardverkiegészítőket ajánlja a COMMODORE-tulajdonosoknak:

	Ára (Ft)
Halász Árpád: Alapismeretek	
a C–64 használatához	140
SFD 1001 felhasználói kézikönyve	120
1526 nyomtató felhasználói kézikönyve	110
MPS 801 nyomtató felhasználói kézikönyve	110
VC 1541 drive felhasználói kézikönyve	110
Dr. Bakó András: Alkalmazói software-k	105
C–600-as kezelési segédlet	240
MPP–1361 nyomtató felhasználói kézikönyve	120
8250 drive felhasználói kézikönyve	183
MS–ADEL felhasználói kézikönyve	2271
MS–SZÖSZI felhasználói kézikönyv	2119
C–600-as számítógép kezelési utasítása	1 968
MS–BASIC felhasználói kézikönyv	1 136
VIC–20 felhasználó kézikönyv	235
Easy Script felhasználói kézikönyv	220
C–64 felhasználási kézikönyv	194
Help-kártya	9 100
80 karakteres screen expander	13 200
Nemzetközi 10-es billentyűzet	6 500
Joystick	2 500
V–24 interface	18 100
Magyar betűkészlet	14 900
Adatmagnó	4 980
Profi copy	6 000
Kommukációs adapter Half	4 900
Full	6 200
Eprom égető	
Fényceruza	
Monitorállvány	427
Porvédő géptakaró mérettől függő	
Antennakábel	250
Csatlakozózsinór	285
Memóriabővítő	38 400
Disc album	153
Cartridge expander	19 800
4 terminálos C–64-esekből álló csoportos	
adatrögzítő	96 000
Inas	7 300
Utility cartridge	9 100
„PARTNER" címlista	
„DESIGN-"	
VEGA	19 100
„TUDOR"	
Telefonkönyv	4 600
„MATSTAT"	14 700
Jogszabály belső	4 900
Jogszabály külső	3 600
Aszinkron intelligensterminál emulátor	28 000
Lineáris programozás VC 1541	19 800
„NOVOCOM"	14 000
MÁTRIX–64	9 500
Sprite Editor	14 200
Képernyő- és táblázatszerkesztő program	4 200
Gyors file loader	10 000

	Ára (Ft)
Sprite karakter Help	9 000
„FORTH–TESTER"	15 000
M Forth a C	30 000
Döntéselőkészítés	11 700
MESTER I.	9 900
MESTER II.	6 600
COPY	4 200
Sort-Merge	4 400
„FES"	19 400
Datapoint emulátor	14 200
Siemens emulátor	
Datafix	17 500
„IEC-BUS" vezérlőrendszer	27 500
Cross +	4 200
Mágneskazettás file-kezelő	4 000
„SEGÉD" filekezelő rendszer	15 000
Matstat szubrutin gyűjtemény	14 700
Nagyító	5 990
Info-sztár	15 900
Lexikon	9 800
Grafikus hardcopy	3 400
Statisztikai nyilvántartó program	10 000
„MANAGER"	8 900
Grafikus programcsomag fényceruza	
bemenettel	19 500
Kézirat-katalógus	6 900
Databasic–64	19 000
Lemezkatalógus-készítő program	3 100
„NOVOTRACE" (monitor trace)	9 800
DIGITEXT	9 500
Írógépelést oktató program	8 500
ISES	37 500
Ékezetes betűk program	12 000
Épületgépészeti tervező program	12 000
Konfekcióipari véghulladék-optimalizáló	
Menetlevél-feldolgozás	30 000
Munkaügyi- és bérfeldolgozási rendszer	26 700
Kiegészítő	8 000
Takarmány-optimalizálás	26 700
Tehénnaptár	26 700
SÍKSTATIKA (mérnöki számítások)	12 300
„COMPOMIX SE"	30 000
Deviza keresztárfolyamot nyilvántartó rendszer	9 400
Műtrágya-szaktanácsadás	28 000
„RANYI" raktári nyilvántartó rendszer	30 000
Áramkör-analízis	17 500
Főkönyvi könyvelés C–64-re	49 000
Villamosenergia-gazdálkodás	14 800
Videókazetta-nyilvántartás	12 500
Árutermelő sertéstelepek	30 000
Keveréktakarmányköltség-minimalizáló	
rendszer	30 000
Termelési program nyilvántartási rendszer	38 000
Személyi nyilvántartás	13 800
Minveszt	12 000

	Ára (Ft)
Állatnyilvántartó rendszer	9 000
Központi jövedelemszabályozás	7 000
Többtelephelyes raktárnyilvántartás	25 000
PM Szüv. – Vezinform	25 000
PM Szüv. – Készáru-nyilvántartás	30 000
PM Szüv. – Anyagnyilvántartás	30 000
PM Szüv. – Havidíjas bérelszámolás	30 000
PM Szüv. – Munkaügyi nyilvántartás	20 000
PM Szüv. – Személyzeti nyilvántartás	20 000
Útvonaltervező program	20 000
Szolgáltatási árvetés nyilvántartó rendszer	9 800
Gyártási önköltség-ellenőrzés	19 000
Mérleg-ellenőrző program	
Biankó tesztprogram	5 000
Melegszalag hengerlés	50 000
Többtelephelyes anyagkönyvelés	19 800
„BIZAR"	7 100
PRODINFO–'85	130 000
TECHNOR" gépészeti normakészítő	30 000
ALLOC–64 állóeszköz-nyilvántartás	26 400
Adók	8 500
Számlakészítő program VC 1541	16 700
Terv – 85	12 000
Árjegyzékkészítő program	
Személyi és munkaügyi adatbank	
Technor	30 000
Számlakészítő program SFD 1001	28 900
Lineáris programozás SFD 1001	
Nebuló	7 000
Játssz és tanulj angolul	8 500
JÁTÉK–I.	6 200
JÁTÉK–II.	4 300
JÁTÉK–III.	5 000
Alkatrészek tűréstechnikai rendszere	6 900
Alkatrészek illesztésrendszere	6 950
Játékkazetták	
MS–ADEL	85 000
MS-BASIC	76 000
MS-SZÖSZI	75 000
Vevő-szállító nyilvántartó rendszer	58 000
Anyagnyilvántartás	70 000
NOVOMATSTAT	19 800
Iktató	45 000
BASIC bővítés	16 000
Alapsoftware-csomag	
(MS-ADEL, MS-BASIC, SZÖSZI)	200 000
Személyzeti nyilvántartás	81 500
Külkereskedelmi vállalat pénzügyi rendszere	125 000
Vezinform	45 000
Munkaügyi nyilvántartás	93 000
RANYI	75 000
Bérelszámolás	92 600
Főkönyvi és folyószámla-könyvelés	89 000
Párbeszédes lineáris programozás	

Amennyiben valamelyik termékünk megnyerte az Önök tetszését, illetve bővebb tájékoztatást kérnek, kérjük, keressék fel irodánkat!

Cím: Budapest XIII., Kresz Géza u. 14. Tel.: 122-047, 122-095. Telex: 22-7673

VC 20



A Vallató másfél éves fennállása alatt többször kacérkodtunk a gondolattal, hogy a VC 20-at is kínpadra kellene vonni. Ám amikor rovatunk beindult, a VC 20 már akkor is régi gépnek számított; fontosabbak, izgalmasabbak voltak már a piacon. Lényegében le is mondtunk arról, hogy vallassuk. A tavalyi év nagy nyugat-európai árleszállításai azonban megváltoztatták eredeti szándékunkat. Minthogy a VC 20 kifutóban levő konstrukció, gyártását leállították, azonban a meglevő raktári készletek nagysága miatt, olyan áron kínálták Nyugat-Európában a gépet, hogy egyszerre Magyarországon is megsokadosott. Innen is, onnan is, ismerősöktől és ismeretlenektől hallottuk, hogy inkább úgy döntöttek Londonban vagy Hamburgban, hogy nem mennek el egyszer a színházba, helyette hoznak haza egy VC 20-at. Így – sok évvel a gép megjelenése után – itthon is olyan elterjedtté vált, hogy mégis érdemes volt kissé megkésve is vallatni.

A KÍNRENDSZERRŐL

A legősibb, eredetileg kidolgozott kínrendszerünket használtuk, inkvizítoraink úgy ítélték, hogy semmi újdonságra nincsen szükség. Talán nem véletlen ez, hiszen egy 1981-ben piacra dobott számítógépről van szó, amely így belefér az általunk régebben konstruált „skatulyába" is. A gépet akkor, egy júniusi Commodore Show-n mutatták be nagy szenzációként, ára kezdetben 200 angol font volt. (Összehasonlításképpen: ma egy 48 kbyte-os Spectrum is csak majdnem a fele!)

GYÁRI ADATOK

Memóriaméret: 3,5 kbyte
Méret: 4,5x21x7 cm
Súly: 211 dkg

Ár – 2,4

1 KÍN

A szokásos magyarázattal kell kezdeni: itthoni vagy nyugati árról van-e szó? Hát, természetesen az itthoni érte el ezt a megtisztelő kettest. Nyugati piacon az ár gyorsan haladt a megfelelő irányba, lefelé, ma, lényegében a végkiárusítás idején a gép ára kb. 40 angol fontra becsülhető. Azért ilyen rejtélyes a megfogalmazás, mert ennyiért még nem lehet megkapni. Annak a csomagnak az ára 100 font, amely tartalmazza az alapgépet, a hozzá szükséges magnetofont, szoftverfüzetet, néhány kész programot. Azon a piacon kizárólag így adható el még néhány darab, így is inkább a mi irányunkból érkező turisták részére. Ez a „keleti" vásárlási kedv érthető is, mert a hazai piacon – árváltozások ide vagy oda – hosszú éveken keresztül semmi sem történt. Az árcsökkenés „nem gyűrűzött be", ahogyan az olajárrobbanást nem tudtuk megakadályozni, úgy a VC 20 olcsóbbodását igen, a bizományi 50 ezer forintért kínálta akkor is, amikor kint már fillérekbe került. Kétségtelen, hogy porosodtak is a gépek a polcokon és csak 1984 nyarán történt meg az itthoni leárazásuk. Ma a kereskedelmi ár 18–20 ezer forint, amiben sajnos nincsen benne a magnetofon, ami viszont külön 10–12 ezerbe kerül. Így a végösszeg nem marad el egy kisebb Spectrum árától, ami talán igazolja, hogy inkvizítoraink miért tartják még ezt is meglehetősen magasnak. Volt, aki így fogalmazott: „csak azért nem egyes, mert lehet kapni!"

Perifériák – 4,1

2 KÍN

Ennél a kínnál érzékelhető pontosan a régi, sokat használt konstrukció előnye. Szinte mindenféle periféria kapcsolható és kapható a géphez (külön kérdés, hogy az itthoni árak ebben is meglehetősen magasak). Létezik joystick, fényceruza, AD konverter, RS232 kimenet, nem szabványos és bus csatlakozó, ami azt jelenti, hogy lemezmeghajtó, printer és tulajdonképpen bármi kapcsolható az alapgéphez. Egyik inkvizítorunknak az a véleménye, hogy a Commodore cég későbbi sikerét éppen a sokoldalúan, előrelátóan tervezett gép alapozta meg.

Képernyőkezelés – 3,2

3 KÍN

Először az adatok: színes, félgrafikus képernyő, nyolc keret- és karakterszínt, valamint 15 papírszínt tud generálni. Egy sorban összesen 23 karakter fér el, ami bizony okoz problémákat. A grafikus karakterei szépek, a finomgrafikai megoldások azonban kizárólag POKE utasí-

VC-20 VALLATASANAK EREDMENYE
1985. JANUAR 10.

K I N O K	TAMASI ANDRAS UZEMELTETO (1)	BRANYI LASZLO FOISK. HALLGATO (2)	GELLERI PETER ALT. ISK. DIAK (3)	KADAS GABOR PROGRAMOZO (4)	LANCSAK ZOLTAN EGYETEMI HALLG. (5)	SZVETNIK ENDRE GIMN. DIAK (6)	DR. SZEKELY JENO FOISK. DOCENS (7)	SZABO CSABA SZAKKOZEP DIAK (8)	ATLAG
1 .KIN:AR	3	2/3	3	1	2	2	3	3	2.4
2 .KIN:PERIFERIAK	4	4	3	5	3	5	4/5	4	4.1
3 .KIN:KEPERNYOKEZELES	3	3	3/4	3	3	3/4	3	4	3.2
4 .KIN:HANG	4	4	-	4	4	3/4	4	4	3.9
5 .KIN:KAZETTAS TAROLAS	5	4/5	4	5	4	5	2/3	3	4.1
6 .KIN:GEPI KODU PROGRAMOZAS	4	2/3	-	1	-	2	-	4	2.7
7 .KIN:MEGBIZHATOSAG	-	4	3/4	5	3	5	5	5	4.4
8 .KIN:BILLENTYUZET	5	4	4/5	5	4	5	4	4	4.4
9 .KIN:DOKUMENTACIO	3	2/3	3/4	2	-	2	3	4	2.9
10 .KIN:EDITALAS	5	4	4	3	3	4	4	5	4.0
11 .KIN:A GEP PROGRAMNYELVE	3	3	3	3	3	2	2	3	2.7
12 .KIN:TANULHATOSAG	-	3/4	3/4	4	3/4	4	3	3	3.5
13 .KIN:EMBERKOZELSEG	5	3/4	3/4	4	4	3	3	3	3.6
+ 1 KIN:SZUBJEKTIV VELEMENY	5	3	3	5	3/4	3/4	3	3/4	3.7
ATLAG	4.1	3.4	3.5	3.6	3.3	3.5	3.4	3.7	3.6

tásokkal, hosszabb bíbelődés árán érhetők el. A vélemény lényegében egyöntetű: nem is lenne olyan rossz, ha egyszerűbb lenne. De mivelhogy nem egyszerűbb... Ezt inkvizítoraink közül valaki így fogalmazta meg: a gép grafikai megoldása sok szempontból hátrányos, ezzel szemben lassú.

Hang – 3,9

4. KÍN

Általánosságban a hangra is valami hasonló igaz, mint ami a képernyőkezelésnél megfogalmazódott: tulajdonképpen sokfélét tud, csak túl bonyolult, és helyenként hibás. Érdekes, hogy talán ezeknél a tulajdonságoknál érzékelhető a legjobban, hogy ez a konstrukció még a kezdetekből származik, így néhány olyan tulajdonságát, amit akkor lelkesen üdvözöltünk, ma már hibának tartjuk – hiszen azóta szellemesebb, egyszerűbb, jobb megoldások születtek ugyanarra. Három hang- és egy zajcsatorna van a VC 20-ban, programozásuk kissé bonyolult, a hang a televízió hangszóróján szólal meg. Többen megjegyezték, hogy – mivel finoman nem szabályozható – gyakran ad hamis hangokat is. Hát hiszen, ha ezt Johann Sebastian tudta volna... Néhány célzás szólt arról is, hogy Junoszty típusú televízión ritkán szólal meg, de ez nem biztos, hogy a VC 20-as hibája.

Kazettás tárolás – 4,1

5. KÍN

A géphez gyári magnetofon készült, amely nem helyettesíthető más, a kereskedelemben kapható kazettás magnetofonnal. Ennek az oka, hogy a magnetofon maga formálja a jelalakot, amit más készülékek nem tudnak. Igaz, kapható, létezik olyan kiegészítő kapcsolás, amivel ez megoldható, erről azonban inkvizítorainknak nincsen információjuk. A gyári magnetofonról viszont szinte egyöntetű volt a vélemény, hogy strapabíró, megbízható. Csak az ára borsos kissé. A gép rendelkezik FILE-kezelő utasításokkal is, kezelés közben a képernyőn jelzi a magnó vezérlését és programellenőrzést is végez. Az egyetlen komolyabb hibája, hogy lassú.

Gépi kódú programozás – 2,7

6. KÍN

Inkvizítoraink közül nem mindenki osztályozta ezt a kínt, ugyanis többen nem használták gépi kódban. Ennek az oka, nemcsak inkvizítoraink lustaságában keresendő, hanem abban is, hogy mazochista legyen a talpán, aki a VC 20-at gépi kódban használja. Egyik inkvizítorunk szerint az alapgépet memóriabővítés és egyéb trükkök nélkül szinte lehetetlen így programozni. Jóleső hümmögés, bólogatás volt a többiek reakciója erre a véleményte, mint-hogy kiderült az, hogy nemcsak nekik nem sikerült. Egyetlen inkvizítorunk írt olyan segédprogramot, amellyel – bevallása szerint a gépi kódú programozás: „viszonylag egyszerű". Nem ellenőriztük.

Megbizhatóság – 4,4

7. KÍN

Általános vélemény, amit az osztályzat is igazol, hogy a gép megbízható, nemigen van vele probléma. Néha melegszik, de azért dolgozni hajlandó, a billentyűzettel sem volt baja még senkinek. Egy inkvizítorunk mondta csak, hogy néha elszáll a gép, és semmit sem hajlandó csinálni, de ez tipikusan az az eset, hogy ha a boltban ki lehetne cserélni egy gyári hibás készüléket, akkor nem lenne probléma. Csak az a bolt egy kicsit messze van. Felmerült az is, amit a Commodore 64-nél is megállapítottunk, hogy a tápegység kizárólag robbantással javítható, mert egybeöntött műanyag házba szerelték.



Billentyűzet – 4,4

8 KÍN

Van akinek tetszik a formája és a színe, van akinek nem – de tulajdonképpen mindenki elégedett vele. Még az az inkvizítorunk is, aki saját véleménye szerint meglehetősen érzékeny a jó billentyűzetre.

Dokumentáció – 2,9

9 KÍN

A Commodore 64-nél leírtuk a gyártó cég véleményét: „kérem, mi gépet gyártunk, nem pedig dokumentációt!" A megállapítás a VC 20-ra fokozottan igaz: a gépkönyv kevés, gyenge, helyenként hibás is, ráadásul magyar fordítása egyelőre nem jelent meg. Így inkvizítoraink ki-ki vérmérséklete szerint osztályozott, nyilván, akik már előzőleg dolgoztak 64-esen, azok mérgesebben. Van, aki szűkszavúan csak ennyit írt: „szép képek vannak benne". Hát, nem nagy dicséret.

Editálás – 4,0

10 KÍN

A gép editora azonos a C 64-esével, tehát teljes képernyő editálására van lehetősége (FULL SCREEN EDITOR). Ahogyan az osztályzat mutatja, ezzel alapvetően inkvizítoraink elégedettek, inkább csak néhány (gyanítjuk, Sinclair-hivő) vallja, hogy kissé bonyolult.

A gép programnyelve – 2,7

11 KÍN

Hát nem valami hízelgő az osztályzat a Commodore cégnek. Hiányzik az ELSE, a SET, és RESET (pontkigyújtás), hiányzik a PRINT AT és meglehetősen sok funkciót csak POKE utasításokkal lehet elérni. Elkényeztetett programírói lelkünknek ez már egy kicsit sok. Vagyis kevés.

Tanulhatóság – 3,5

12 KÍN

Lényegében ebben a kínban összefutnak a már említett különböző kisebb-nagyobb problémák. A bonyolult javítási lehetőség (editálás), a POKE utasítások használata, a rossz dokumentáció együtt eredményezik, hogy inkvizítoraink véleménye szerint nem könnyű a gép birtokbavétele. Igaz, valamenyien hozzátették, hogy „végül is sikerült, sőt megszerettem", de azt is mindenki bevallotta, hogy nem ez volt az első gépe az életben. Márpedig amit gyakorlott programozó kis nehézségekkel sajátít el, az egy kezdő számára vér-verejtékes küzdelem. Ahhoz képest, hogy a VC 20 akkoriban kifejezetten kezdőknek készült, nem nevezhető sikeres konstrukciónak.

Emberközelség – 3,6

13 KÍN

Szinte folytatni lehetne az előző gondolatsort, a konstrukciós kiforratlanságok miatt az osztályzat nem igazán jó. Itt általában a kis memóriát és a BASIC hibáit említették újra, amiből talán az érzékelhető, hogy a VC 20 szinte kísérlet volt a 64-es előtt, hogy mit hogyan kell csinálni. Így azután nem minden sikerült tökéletesen, de azóta volt idő kijavítani.

Szubjektív vélemény – 3,7

+1 KÍN

Az előzőek összegzése az általános vélemény, amelynek osztályzata meglepően rossz ahhoz képest, hogy ugyanakkor hányan mondták, írták azt, hogy végül is megszerették a gépet. Ennek az ellentmondásnak talán az lehet a magyarázata, hogy a még nem teljesen kiforrott konstrukció azért nem sikerült rosszul, bizonyítéka, hogy mennyi minden megmaradt belőle még a Commodore 64-esen is. Így tehát meg lehet kedvelni a gépet, de egy inkvizítor, ha egy kicsit ad magára, akkor igazán jó osztályzatot már nem ad egy ősleményre. Így talán a VC 20-as lesz az egyike azoknak az első gépeknek, amelyet – jóleső érzéssel – hamarosan elfelejtünk...

HOZZÁSZÓLÁS

Tisztelt Szerkesztőség!

Önök két öngólt rúgtak a VIC–20-as számítógéppel kapcsolatban! Az egyik az, hogy csak az „erőszaknak engedve", éves késéssel tűzték napirendre a gép tesztelését. A másik az, hogy ÍGY!
Nem akarom megbántani a Vallatóban résztvevőket, de a többségükről a véleményük (tévedéseik) alapján kitűnik, hogy néhány óránál többet nem ültek a VIC–20 előtt. Ilyen mélységű ismeretek alapján pedig nem illik megalapozott véleményt mondani egy gépről (aki nem tud arabusul...).
Felsorolnék néhány dolgot, amely kétségessé teszi a kapott osztályzatok egy részének megalapozott voltát.
1. kín: Miért változott meg az eddigi gyakorlat, hogy a külföldi árat osztályozzák? A vallatás ezzel a nem teljesen fair változtatással kezdődik.
(Ennek oka egyszerű. A legutóbbi nyugati leértékelés után olyan olló keletkezett a kinti és itthoni ár között, amely irreálissá tette volna a kinti végkiárusítási ár osztályzását. S egyáltalán nem reális egy ilyen dömpingárat osztályozni, amely nem tükrözi a gép kinti árfekvését sem! – A szerk.)
2. kín: Ha a géphez minden periféria kapható (és kapható!), akkor miért kapott a gép hét 3-ast is?
3. kín: A leírtakkal ellentétben 22 karakter írható egy sorba! A Super Expander nevű programmal BASIC-ből is egyszerűen kezelhető a nagy felbontású grafika!
4. kín: A Super Expanderrel a hanggenerátor is egyszerűen kezelhető. A Junoszty tv-n a hang soha nem is fog megszólalni rendesen, mivel a gép modulátora PAL rendszerű, ezért csak kétnormás (PAL-SECAM) tv-vel kapunk színes képet és hangot! Ezt még egy Vallatónak is illene tudni!
5. kín: A kazettás tárolás sebessége

SZVETNIK ENDRE: **Könnyen kezelhető, bár editálásnál bele lehet gabalyodni.**

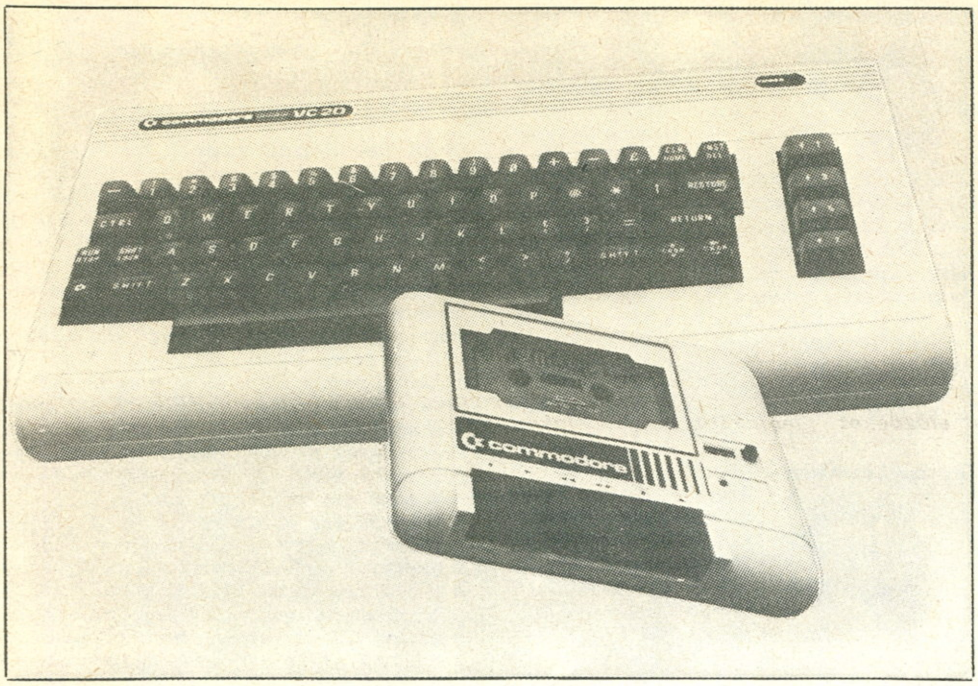
SZÉKELY JENŐ: **A kezdőket gyakran becsapja.**

TAMÁSI ANDRÁS: **Jó kazettával csak én hibáztam.**

LANCSÁK ZOLTÁN: **Teljesen megbízható, bár egy kicsit melegszik.**

GELLÉRI PÉTER: **A formatervezés kitűnő, de a tudása már nem annyira.**





ugyanakkora mint a piacon kapható gépek 99%-ának! De ha ez valakinek nem elég, kapható olyan cartridge, amely ezt a sebességet a tízszeresére növeli! A kazettás tárolást huzamosabb ideje használók sem igen tudják, mi az, hogy LOAD ERROR! Valószínű, hogy a SPECTRUM, PRIMO, HT stb. felhasználói irigykedve sóhajtanak fel ennek hallatán!

6. kín: A hümmögés inkább azért lehetett, mert a Vallatók nem tudták, hogyan lehet gépi kódban programozni a VIC–20-at. Mindenféle tárbővítés nélkül a VICMON nevezetű program segítségével nagyszerűen lehet gépi kódban programozni a gépet! Méghozzá úgy, hogy a bebillentyűzött mnemonikot rögtön ellenőrzi, lefordítja és elhelyezi a memóriában. Ideális eszköz egy kezdőnek, hogy megtanuljon gépi kódban programozni!

7. kín: A tápegység a „buherálók" ellen, életvédelmi szempontból van műgyantával kiöntve. Egyébként is miért kellene javítani? Nem szokott elromlani!

8. kín: A billentyűzetre kapott osztályzattal kapcsolatban csak annyit, hogy aki ebben a kategóriában ennél jobbat akar, annak azt kívánom, hogy egész életében csak ZX81 billentyűzetet nyomkodjon!

9. kín: A felhasználói kézikönyv valóban nem túl bőbeszédű, viszont létezik a VIC–20 programozói kézikönyv, amely sokkal részletesebb, megadja a memóriatérképet, sőt jó néhány interpreter rutin leírását, kezdőcímét is megadja, példákkal illusztrálva a használatot.

10. kín: Aki dolgozott már SCREEN EDITOR-os géppel, az tudja, hogy ennél kényelmesebb mód a szerkesztésre nincs. Nem éppen hízelgő a vallatókra, hogy a 4 (négy!) editorgomb használata tábba belebonyolódtak!

11. kín: A gép programnyelve teljes egészében megegyezik a Commodore 64 gépével, ezért kicsit csodálkozom, hogy ennyire rossznak találtatott. Sajnos nem tudtam rájönni arra, mit takar az a homályos utalás, hogy „meglehetősen sok funkciót csak POKE utasításokkal lehet elérni". Az interpreterről csak annyit, hogy az ilyen egyszerű és olcsó gépeknél nem teljesen tipikus például a file-kezelésnek az a széles köre, amit a VIC–20 tud (kazettás, disces, RS–232). Egyébként aki az eredeti interpreternél jobbat akar, annak rendelkezésére áll két olyan program (cartidge), amely kibővíti az utasítás készletét. Ez a PROGRAMMER'S AID és a SUPER EXPANDER.

12. kín: Az egész teszthez való hozzáállás és „hozzáértés" tükröződik ebben a pontban. A C 64-gyel teljesen megegyező editor bonyolultnak van titulálva, és állítólag „vér-verítékes küzdelem" kell a gép megismeréséhez. A vallatók okosabbak maradtak volna, ha nem vallatnak.

13. kín: A VIC–20-szal teljesen megegyező BASIC-kel és eidtálási lehetőségekkel rendelkező C 64-et annak idején a szakemberekből álló vallatók „túlságosan is emberközeli"-nek minősítették. Hát erről ennyit!

+1 kín: Az én kínom az, hogy nemcsak ezt az egy pontot értékelték szubjektíven a vallatók, hanem a többi 13-at is! Ha a VIC 20 azokra a funkciókra, amelyek teljesen azonosak a C 64-gyel, ugyanazt az osztályzatot kapná, mint annak idején a C64, akkor az átlaga máris 4,0 lenne, és akkor még az árról nem is beszéltünk!

Összegezve az eddigieket, az én szubjektív véleményem az, hogy **ne a VIC–20-at felejtsük el, hanem a Vallatót!** Minden eddigi morgások ellenére maradok a BIT-LET hűséges olvasója.

Szilvágyi Gábor 2092 Budakeszi, Arany J. u. 7. fsz. 3.

Az Ötlet 1985. március 28-i számában megjelent VC 20 vallatóhoz szeretnék hozzászólni. Szerintem több kínnál is hátránynak vették azt, amit a C 64-nél előnyként soroltak fel. (Nem voltam rest, és elővettem a C 64 vallatóját.)

Az alábbiakban felsorolom azokat a kínokat, melyekkel nem értek egyet (bár ez nem sokat jelent).

2. kín: perifériák. Ebben Önök is elismerik, hogy a VC 20-hoz minden csatolható és kapható, csakúgy, mint a C 64-hez. Ebben a két gép megegyezik, mégis a VC 20 8 tizeddel kevesebbet kapott. Miért?

3. kín: képernyőkezelés. Tényleg nem kiváló. Egyébként azt írják, hogy finomgrafika „kizárólag POKE utasításokkal" érhető el. A C 64 alapgépen nem? Különben a kizárólagosságot is megkérdőjelezem, ugyanis a SUPER bővítő modullal (VC 1211 A)160x160-as felbontás érhető el, és egyszerűen kezelhető grafikai utasításokat is tud (pontrajzolás törlés, körrajzolás, egyenesrajzolás-törlés, kifestés stb.). Az azonban igaz, hogy a C 64 hardver tekintetében kiforrottabb, többet tud. Ha a C 64-nél is az alapgépet osztályozták, hogy kaphatott 4,8-at?

Ha pedig azt bővítéssel vallatták (SIMON'S BASIC, SUPERGRAPHIC), akkor a VC 20-nál miért nem nézték meg a bővítéseket?

5. kín: kazettás tárolás. Ehhez csak annyit, hogy lassú (de kapható hozzá TURBO-modul, ami a 10-szeresére (!) gyorsítja a felvételeket, beolvasásokat), az azonban nem igaz, hogy a gyári magnó mással nem helyettesíthető. Mi két és fél éve SANYO-val használjuk, megbízhatóan. A discet pedig éppúgy kezeli, mint a C 64.

6. kín: gépi kódú programozás. Ennél a kínnál az inkvizítorok azt osztályozták, hogy milyen fordítóprogramok kaphatók a géphez. (Egyébként a VC 20-hoz is lehet jót kapni!) Hogy lehet 1,4 jegy különbség a két processzor gépi kódú programozásában, holott azok szoftverkompatibilisek? Egy apróság: a C 64 alapgépet mennyivel könnyebb gépi kódban programozni?

7. kín: megbízhatóság. Miben megbízhatatlanabb a C 64-nél?

8. kín: billentyűzet. Ez bosszantott fel a legjobban. A C 64-nél csillagos ötöst akartak adni, a VC 20 4,4-et kapott, pedig a billentyűzetük egy az egyben ugyanaz! UGYANAZ!

10. kín: editálás. Önök is elismerik, hogy editora azonos a C 64-esével. Akkor miért kapott 4 tizeddel kevesebbet? És ki bonyolódhat bele? Ennél egyszerűbb, jobb javítási lehetőség nincs.

11. kín: a gép programnyelve. Hát ez tényleg nem jó! És a C 64 alapgépé? A programnyelvük azonos, mégis a VC 20 1,2-del rosszabbat kapott a C 64-nél! Miért?

12. kín: tanulhatóság. A C 64-nél írják: „...a gép felépítése, kezelése valóban olyan logikus és egyszerű, hogy még megfelelő használati utasítás nélkül is megérthető". A VC 20-ra ez nem igaz? És miért bonyolult a javítási lehetőség? (Erről jut eszembe: ha ez bonyolult, akkor a HT 1080Z javítási lehetősége milyen?)

Az előzőekben felsorolt ellenvetésekkel nem azt akarom mondani, hogy a VC 20 olyan, mint a C 64. Nem, erről szó sincs! De ami a két gépen teljesen egyforma, azt miért nem egyformán osztályozták? Még akkor is, ha az inkvizítorok mások voltak!

Pintér Károly 4. osztályos tanuló 8000 Székesfehérvár, József Attila út 59.

A vádak súlyosak. Azonos dolgokat másként osztályoztak az inkvizítorok a C 64-en és a VC 20-on. Nos, mindenre nincs mentség. De az igazsághoz hozzátartozik, kedves olvasónk, hogy az azonosságok – s ez a leveléből

is kiderül – itt-ott csak majdnem azonosságok. A válasz másik része pedig ott van az Ön levelének utolsó mondatában. Az inkvizítorok nem mindig ugyanazok, ezért az osztályzatok is hol szigorúbbak, hol enyhébbek. Sajnos ez Vallató rovatunk gyengéje. Sajnáljuk, mégis szeretjük ezt a rovatot betegségeivel, korlátaival együtt. De levele alapján úgy vesszük észre, bármilyen mérges most, azért titkon Ön is szereti a Vallatót!

Mivel VC 20 tulajdonos vagyok, lehetetlen, hogy tollat ne ragadjak vallatásának hatására.

Először is a hanggal kapcsolatban van egy észrevételem. A Junoszty televízió áthangolásával kiválóan szól azon is a VC 20. Sajnos ehhez bele kell nyúlni a televízióba (szakembernek), mivel azt nem Commodore gépekhez tervezték.

Másodszor a gépi kódú programozásával kapcsolatos felületes vallatásukhoz kívánok szólni Ez ügyben összehasonlítom a Z 80 gépi kódú programozásával.

1. Eleve egyszerűsíti a helyzetet, hogy a processzornak a veremtármutatón és programszámlálón kívül csak egy akkumulátora és két indexregisztere (XJ) van, továbbá a feltételregiszter, szemben a Z 80 regiszterkészletének tömegével (A, F, B, C, D, E, H, L, a másodregiszterek IX, IY).

Ami azonban nem teszi ügyetlenebbé a gépet, mivel a használható zéró-lapos címzési forma lehetővé teszi, hogy mindegyik 0-lapos memóriacímet REGISZTERKÉNT kezeljük, és így tulajdonképpen 256 (!) plusz regiszter birtokosainak érezhetjük magunkat!

2. A processzornak jóval kevesebb az utasításkészlete mint a Z 80 processzornak (152 db ismert utasítás + 32 db ismeretlen, szemben a Z 80 több mint 700 utasításával), ennélfogva könnyebben áttekinthető és megjegyezhető.

Sőt! Míg a Z 80 műveletet csak az akkumulátorral, a regiszterekkel a HL, IX, IY regiszterpárokkal képes végezni, a mi processzorunk a memória összes byte-jával képes műveletet végezni. Például ha a Z 80-nál az akkumulátorhoz hozzá akarjuk adni egy cím tartalmát, akkor először betöltjük a címet a HL regiszterpárba, majd következő utasításban hozzáadatjuk az akkumulátorhoz a HL által mutatott cím tartalmát. Mi azonban egyetlen utasításban hozzá tudjuk adni az akkumulátorhoz bármelyik cím tartalmát

Z 80 LD HL, cím 6502: ADC cím
ADC A,(HL)

Így a Z 80-nál szükségessé válik a regiszterekbe töltés – és a regiszterek közötti csereutasítások nagy számú megnövelése. Vagy itt vannak a Z 80 16 bites regiszterpárjai, amit mi kiválóan helyettesíthetünk egy zérólapos címzéssel.

Például hasonlítsuk össze az akkumulátor tartalmát az IX regiszterpár és egy d eltolódás által mutatott cím tartalmával a Z 80 esetében:

CP (IX+d) (3 byte hosszú)

És hasonlítsuk össze a 6502 esetében:

CMP (z)+y (2 byte hosszú)

ahol z egy két byte-os cím első byte-jára mutat a zérólapon és az eltolódást az Y regiszter tartalma jelenti. Sőt, míg a Z 80-nál csak IX-szel tudjuk módosítani a zérólapos cím mindkét byte-ját, és az eltolódást jelző Y regisztert is!

Így valójában úgy vetődik fel a kérdés, hogy a kétféle processzor közül melyik tud több 16 bites regiszterpárt használni. A Z 80 használja a BC, DE, HL, IX, IY regiszterpárt, a 6502 pedig felhasználhatja a zérólap 256 byte-ját „regiszterpárként"! (Azaz a 128-at.)

Nem csoda, ha a Z 80 memóriakezelése szegény, jócskán ki kell bővíteniük utasításkészletüket regiszterekbe töltő és memóriacímre töltő utasításokkal, ezért az a temérdek LD utasítás, annyira, hogy még tömbös töltőutasítás is szükségessé vált!

3. Megírtam egy DISASSEMBLER és egy ASSEMBLER programot VC 20-ra gépi kódban. Segítségükkel ezerszeresen egyszerűsödött maguknak a kódoknak a fordítása is. Mivel a programok kb. 1–1 kbyte-osok, ezért legalább 1 kbyte-os gépi kódú anyagot tudok egyszerre megírni a VC 20-ra!

4. Aki azt mondta, hogy a gépet memóriabővítés nélkül lehetetlen gépi kódban programozni, az nyilvánvalóan nem is értett hozzá. Amióta birtokomban van, memóriabővítés nélkül programozom gépi kódban.

Amiképpen lehet még az 1 K-s ZX81-et is gépi kódban programozni memóriabővítő nélkül, annál inkább lehet a VC 20-at. Elterjedt programozási forma, hogy gépi kódú anyagunkat egy BASIC sorban a REM utasításban tároljuk, kikeressük, hogy melyik memóriacímre helyezi az INTERPRETER, és onnan hívjuk meg gépi kódú szubrutinunkat. Másik forma, hogy DATA-ban tároljuk és programfutáskor kiküldjük az adatokat egy letiltott területre, ahol az INTERPRETER nem nyúl hozzá. Az előző formátum hátránya, hogy ki kell keresnünk a memóriából, hogy az INTERPRETER hová pakolta sorunkat, és így megállapítani, hogy melyik címet hívjuk meg, ami már kettőnél több gépi kódú szubrutin írásánál komoly gonddal és vesződséggel jár. A második formátum hátránya pedig a nagy helyfoglalás. Egyszer a DATA soraink foglalnak le a memóriában területet, másodszor pedig le kell tiltanunk egyéb szabad területet is, ahová küldjük a gépi kódú anyagunkat programfuttatáskor. A VC 20-nál mindkét formátum adott. Azonban én ajánlok két egyszerűbb és könnyen kezelhető programot, amellyel gépi kódú anyagunkat kazettára vihetjük és onnan pedig tetszőleges helyre beírhatjuk.

A lenti rövid programot futtatva először a „PRESS RECORD & PLAY ON TAPE" üzenet jelenik meg. Nyomjuk meg mindkét billentyűt a kazettás magnón. Mikor megáll a magnó, a képernyőn a kérdőjel jelenik meg, várva a gépi kódú adatokat. Azokat szép sorjában adjuk be, legutoljára egy nullánál kisebb számot. Ha sok adatot viszünk be, időnként beindul a magnó, és kazettára kerül az anyagunk. Ilyenkor mindig várjuk meg, míg a kérdőjel megjelenik. Amikor beadtuk a legvégét jelző negatív számot, akkor két szám jelenik meg: az első egy ellenőrző szám, a gépi kódú anyagunk adatainak összege, a második pedig az adataink száma. Végül a programunk önmagát kitörli a memóriából.

```
1000 OPEN1,1,1:REM KAZETTAS CSATORNA MEGNYITASA
1001 INPUTA:IFA<0THEN1005:REM ADATOK KERESE,MIG NEGATIV ADAT NEM JON
1002 K=K+A:N=N+1:REM K=ADATOK OSSZEGE,N=ADATOK SZAMA
1003 PRINT#1,A:REM KAZETTARA IRAS
1004 GOTO1001:REM FOLYTATAS
1005 CLOSE1:PRINTK;N:REM KAZETTAS CSATORNA LEZARASA,K;N KIIRASA
1006 NEW:REM A PROGRAM KITORLESE A MEMORIABOL
```

Ha most azt akarjuk, hogy a kazettára mentett gépi kódú anyagunkat a gépbe vigyük, üssük be a következő programot.

```
2000 INPUT"MELYIK CIMTOL";P
2001 POKE56,P/256:POKE55,PAND255:CLR
2002 OPEN1,1,0
2003 P=PEEK(55)+256*PEEK(56)
2004 INPUT#1,A
2005 POKEP,A
2006 K=K+A:N=N+1:P=P+1
2007 IFST=0THEN2004
2008 CLOSE1:PRINTK;N
2009 NEW
```

2000 sor: Tudnunk kell, hogy 3583 byte áll rendelkezésünkre, 4096–7679 címeken. Tehát kezdő címünk bárhol elhelyezkedhet ezen a területen, természetesen hagyjunk helyet BASIC-programunknak is, amely 4096-os címen kezdődik.

2001 sor: Letiltja az INTERPRETER számára azt a területet, ahova anyagunkat helyezni szeretnénk.

2002 sor: Kéri, hogy kapcsoljuk be a PLAY gombbal, megindítja a magnót és beáll a kazettán levő adataink elejére.

2003 sor: A CLR utasítás törölte P értéket, de az 55–56 címen az tárolva van.

2004–2006 Beküldi összes adatunkat a meghatározott címtől kezdődően kazettáról a memóriába, számolja összegüket és darabszámukat. Ezt mindaddig csinálja, míg az adatok vége jelzést nem kapja kazettáról.

2008 Leállítja a magnót, kiírja az ellenőrzőszámot és adataink számát, itt ellenőrizhetjük, hogy rendben ment-e minden.

2009 ez törli BASIC-programunkat a memóriából.

Most tehát betöltöttük gépi kódú anyagunkat a gépbe, tudjuk, hol keressük, a többi terület teljesen szabad, betölthetjük oda azt a BASIC-programot, amivel gépi kódú szubrutinjainkat használni akarjuk. Maga a töltőprogram pedig semmiféle helyet nem foglal el, mivel kitörlődött.

5. Dokumentációval kapcsolatos észrevételem. Valóban csapnivalóan sovány információval látták el a gépet. Ez ügyben felajánlom saját információimat, amiket szereztem a gépről, mivel lefordítottam magamnak a teljes INTERPRETER anyagát.

Végezetül, véleményem szerint a VC 20 gépi kódú programozása nemcsak egyszerű, könnyen elsajátítható, de maga a nyelvezete is rendkívül ötletes. Gépi kódú programjaink némiképp gyorsabbak is lehetnek azáltal, hogy akkumulátorunk közvetlen kapcsolatba kerülhet a memória összes byte-jával, és így nem kell azokat állandóan töltögetni ide vagy oda.

Tóth Kornél 1083 Leonardo da Vinci u. 29. fsz. 8.


VIDEO-HI-FI
SZAKÜZLET
BP. IX. ÜLLŐI ÚT 69.
IBM-XT SZEMÉLYI
SZÁMÍTÓGÉPEK SZÁLLÍTÁSÁT
RÖVID HATÁRIDŐVEL
VÁLLALJUK!
ÜZEMBE HELYEZÉS – SZERVIZ
RAMOVILL
RAMOVILL
COMPUTER
SZAKÜZLET
Bp. IX., Üllői út 69.
Telefon: 343-125
Telex: 22-6127

A Szuper BIT-LET végre lehetőséget teremtett arra, hogy felrúgjunk minden eddigi szabályt és ellentmondjunk saját magunknak. A VALLATÓ rovat ugyanis egyértelműen úgy fogalmazta meg alapelvét, hogy nem akarunk gépeket összehasonlítani, nem akarjuk a ZX 81-et a Commodore 64-hez viszonyítani, a kapott osztályzatok nem fejeznek ki rangsort – hanem kizárólag azt, hogy a saját kategóriájában, a saját lehetőségei között egy gép mit tud. Persze tudtuk, hogy a célkitűzés eleve kudarcra van ítélve, tudtuk, hogy inkvizítoraink nem tudják agyukból kitörölni egy másik gép emlékét és az átlagok kiszámítása után első dolgunk volt, hogy összehasonlítsuk a kapott jegyet előző gépekével. És titkolt reményeink között azért az is szerepelt, hogy olvasóink otthon egymás mellé teszik a Vallató számait és bizony összehasonlítják a gépeket. Igazából magunk is alig vártuk a pillanatot, hogy elkészüljön az a táblázat, amelyikben az összes gép összes osztályzata szerepel. Íme itt van, értelmezéséhez azonban néhány dolgot nem árt tudatosítani.

	VC-20	COMMODORE 64	ZX-81	AIRCOMP	HT-1080 Z	ZX SPECTRUM	ABC-80	PRIMO
1 .KIN AR	2.4	(4.8)	4.2	4.2	3.6	(4.8)	---	4.1
2 .KIN PERIFERIAK	4.1	(4.9)	3.1	2.2	3.4	3.9	3.1	---
3 .KIN KEPERNYOKEZELES	3.2	(4.8)	2.7	4.4	3.2	4.5	3.2	4.3
4 .KIN HANG	3.9	(4.9)	1.6	3.3	4.6	3.6	3.5	3.6
5 .KIN KAZETTAS TAROLAS	4.1	3.8	2.4	3.7	3.6	(4.7)	3.4	3.9
6 .KIN GEPI KODU PROGRAMOZAS	2.7	4.1	2.9	(4.7)	4.5	4.4	3.2	3.2
7 .KIN MEGBIZHATOSAG	4.4	(4.6)	2.5	3.8	4.5	4.4	4.5	3.9
8 .KIN BILLENTYUZET	4.4	(5.0)	2.1	3.1	3.9	3.8	4.9	3.7
9 .KIN DOKUMENTACIO	2.9	2.4	4.3	2.4	2.2	(4.6)	3.2	2.9
10 .KIN EDITALAS	4.0	(4.4)	3.7	(4.4)	3.4	(4.4)	3.9	4.0
11 .KIN A GEP PROGRAMNYELVE	2.7	3.9	3.3	3.8	4.1	(4.6)	4.4	4.1
12 .KIN TANULHATOSAG	3.5	4.6	4.6	4.3	3.5	4.6	(4.8)	4.3
13 .KIN EMBERKOZELSEG	3.6	4.4	4.0	4.3	3.8	(4.9)	4.4	4.0
+1. KIN SZUBJEKTIV VELEMENY	3.7	(5.0)	4.3	4.2	4.0	4.9	4.3	4.0
ATLAG	3.6	4.4	3.2	3.8	3.8	(4.5)	3.9	3.9
+2. KIN SZERVIZ	---	---	---	---	5.0	---	---	---
+3. KIN SZOFTVER ELLATAS	---	4.4	---	---	---	4.9	---	---

1. KÍN: KIK IS AZOK AZ INKVIZÍTOROK?

Ez a táblázat nem tartalmazza az inkvizítorok névsorát, de az előző oldalakról kiderül, hogy az egyes gépeket más és más emberek vallatták. Ez az összehasonlító táblázat akkor lenne megfelelően egyértelmű és szubjektív, ha ugyanazok osztályozták volna az összes gépet. Végül is ez megszervezhető lenne, hiszen néhány szakembert kellett volna csak felkérni arra, hogy foglalkozzanak az egyes gépekkel, és utána adjanak osztályzatokat. Mi azonban inkább olyanokat választottunk inkvizítornak, akik régóta gyűrnek egy-egy típust, így a megbízhatóságáról, tartósságáról és a gép finomságairól is vannak tapasztalataik. Ennek ellenére az a véleményünk, hogy 8–10 ember átlagolt véleménye már megfelelően objektív ahhoz, hogy nem ad hamis képet. Valószínű, hogy más inkvizítorok átlagvéleménye is – néhány tizedes eltéréssel – ugyanezt a sorrendet adná.

2. KÍN: MIFÉLE ÁRRÓL VAN SZÓ?

A Szuper BIT-LET Vallatóiból kihagytuk az árakat, mert az eredeti megjelenés óta tetemesen változhattak. Az osztályzatokat azonban bent hagytuk. Nyilvánvaló, hogy ezek a megjelenés körüli állapotot tükrözik, de most utólag végignézve mégis azt mondjuk, hogy nem irreálisak. Az osztályzatok. Az árak változatlanul irreálisak mindenfajta vám-, jogszabály és árcsökkenés (?) ellenére is. Talán a Primo jelenléte okozhat valami pezsgést az állóárban, de ez egyelőre még nem érzékelhető.

3. KÍN: MÚLIK AZ IDŐ!

Ahogyan inkvizítoraink sem lettek fiatalabbak az elmúlt két évben, ugyanúgy a ZX 81-es konstrukciója sem. Márpedig két év a számítógépek mai fejlődési üteme mellett bizony sokat jelent. Az általunk vallatott nyolc számítógéptípus közül kettő már teljesen eltűnt a piacról (ABC 80, AIRCOMP), lényegében a cégek feladták a küzdelmet, és beszüntették a gyártást. Két másik cég újabb típust dobott a piacra, ezért szüntette meg a gyártást (ZX 81, VIC 20), ezek azonban még kaphatók a piacon. A HT 1080Z abban a kivitelben, ahogyan vallattuk, már nem rendelhető meg, az újabb típust azonban még nem ismerjük. Lényegében változatlanul kapható a piacon három típus – és talán nem véletlen, hogy melyik három: a ZX Spectrum, a Commodore 64 és a Primo.

4. KÍN: A KÍNOK

Tanulságos végignézni a táblázatot soronként és oszloponként. Így utólag is megveregethetik a vállukat inkvizítoraink, nem tévedtek nagyokat; a táblázat lényegében tükrözi a mai technika állását és a világpiaci értékítéletet is. Nem vitás, hogy az általunk vizsgált kategóriában kimagasló két konstrukció a ZX Spectrum és a Commodore 64-es. A Spectrum átlagosztályzata jobb, a Commodore viszont a szubjektív megítélésben abszolút győztes, és több tulajdonságával került az élre. Felújíthatnánk a régi vitát, hogy melyik jobb, de inkább úgy érezzük, a küzdelem még nem dőlt el. Érdekes viszont, hogy miben és mely gépek tudtak betörni a legjobbak közé. Az ABC 80 – tanulhatósága egyértelműen a legjobb. Ezt különösen értékesnek tartjuk, hiszen a gépet éppen erre hozták létre: iskolai, tanuló számítógépnek. Aki tervezett már elektronikus áramkört, az tudja, hogy a számtalan kompromisszum között sokszor éppen a legfontosabb tulajdonság veszít az értékéből – így ez az eredmény nem lebecsülendő. Lelket melengető érzés, hogy egy magyar konstrukció két tulajdonságával is a legjobb lett; lelket fagyasztó, hogy pontosan arról az AIRCOMP-ról van szó, amelyik azóta a piaci harcok következtében kimúlt. Persze a két kín természete egy kicsit az okot is megvilágítja: a gépi kód és az editálás a számítógépek „agyának", működési módszerének, a szoftvernek az elemei. A jó szoftver elkészítéséhez nem kell más, csak néhány jó szakember, akik ismerik a legjobb megoldásokat, és azokon még újítani is tudnak. Csak ez nem elég! A számítógépek gyártásához ipari háttér, alkatrészellátás, képzett szakmunkások kellenek, és reális, a piacon elfogadott ár. Úgy tűnik, az AIRCOMP ezeken bukott el, amivel pontosan modellezte a mai magyarországi ipar helyzetét.

5. KÍN: ELÉGEDETTEK LEHETÜNK?

A Vallató szándéka mindig az volt, hogy orientáljon felhasználót, számítógép-vásárlót, adjon információt arról, hogy az egyes felhasználási területekre melyik a jobban alkalmazható gép. Persze tudjuk, hogy a szakemberek nem tőlünk szerzik az információikat, de reménykedünk abban, hogy rövid idő alatt kialakul Magyarországon az a réteg, amely különböző céllal otthonra vásárol számítógépet. És számukra a Vallató segítség lehet. Minthogy ez a réteg még mindig nagyon szűk, ezt a reményünket még mindig nem veszítettük el.

Pogány György
a Vallató rovat vezetője

SZUPER VALLATÓ

Gyakorlott inkvizítorainknak egy idő után elege lesz ugyanazokból a kínzóeszközökből! Hát még abból, ha mindig ugyanolyan az, aki vagy ami a kínpadon van. Talán ez volt az egyik oka annak, hogy újfajta alany után néztünk. A másik – egy kicsit fontosabb – szempont, hogy lassan elfogynak a Magyarországon pillanatnyilag elismert, elterjedt, vallatásra érdemes mikroszámítógépek. Ugyanakkor végre nagy menynyiségben, már-már elterjedt a Spectrum Microdrive, ez a nagy hírveréssel beharangozott műszaki csoda, amiről többnyire a tulajdonosok is ellentmondó információkkal rendelkeznek. Így különös élvezettel, fogcsattogtatva gyűltünk össze, hogy végre a különböző típusú alapgépek helyett egy kiegészítő berendezést, egy háttérmemóriát vallathassunk.

GYÁRI ADATOK

Jobbára nincsenek. A gépkönyv semmiféle műszaki adatot nem tartalmaz, még a tárolókapacitást sem adja meg, aminek mint később kiderült, oka van, így az itt következő adatok inkább tapasztalati értékek.

Csatlakozási lehetőségei: két csatlakozóval rendelkezik, az egyikbe az Interface 1, a másikba az esetleges második Microdrive kapcsolható.

Memóriakapacitása: az ígért 86 kbyte helyett 94 és 110 kbyte között.

Mérete: 95x85x45 mm

Súlya: 175 gramm (kazetta nélkül)

KÍNRENDSZER

Ez elsősorban nekünk, inkvizítoroknak okozott komoly kínokat. A személyi számítógépeknél jól bevált, már élvezetes kínok egy egészen más rendeltetésű eszköznél használhatatlannak bizonyultak. Kicsit úgy éreztük magunkat, mint azok a középkori inkvizítorok, akik jól bevált sötét sziklabörtönük mélyén egyszer csak azt a feladatot kapták, hogy vallassanak ki egy teknősbékát. A spanyolcsizma erre alkalmatlan. Így elsőként mi is új kínzóeszközök után néztünk. Nem volt könnyű.

Inkvizítoraink első lépésben megegyeztek abban, hogy a Microdrive önmagában nem vallatható, hiszen csak egy rendszer részeként működik. A Microdrive nem több, mint egy magnetofon. Kazetta és erősítő nélkül pedig egy magnó sem szólal meg. Tehát viszonylag gyorsan egyetértettünk abban, hogy a Vallatás csak úgy lehet eredményes, ha a Spectrum-Interface-Microdrive-kazetta rendszert vizsgáljuk, erre dolgozzuk ki a kínokat. Miután ezt közösen eldöntötték inkvizítoraink, rögtön abban is egyetértettek, hogy ez így nem lesz jó, mert bizonyos részekről külön-külön érdemes beszélni. Mire megfelelően összezavartuk egymást, és az addig tisztán látókkal is sikerült elhitetnünk, hogy sötétben tapogatóznak – a következő kínrendszert kínlódtuk ki.

Ára a nyugati piacon – 4,1

1 KÍN

Minthogy a Microdrive közvetlenül nem kapcsolható az alapgéphez, ezért ebben az esetben természetszerű, hogy az Interface 1 és a Microdrive együttes árát vizsgáltuk. A berendezések ára gyakorlatilag a fél évvel ezelőtti megjelenés óta nem sokat változott, jelenleg is 100 angol font körül van. (kb. 50 az Interface, és ugyanannyi a Microdrive). Néhány jól értesült inkvizítor hallott már olyan beszerzési forrásokról, ahol féláron kapható ugyanez, és – bár inkvizítoraink többsége az ilyen mendemondáknak nem hisz –, azért vallatásunk szünetében többen lopva feljegyezték annak a bizonyos álombeli bécsi boltnak a címét. Mindezekkel együtt, mint az a kapott osztályzatból kitűnik, a nyugati árral általában elégedett volt mindenki. A Sinclair cégtől megszokott alacsony árszínvonal nem változott, a berendezések elérhetőek, sőt ahhoz képest, hogy mit nyújtanak, a nyugati piacon olcsónak is ítélhetőek. Az egyetlen hármas osztályzatot adó inkvizítor úgy ítélte meg, a műszaki színvonal alapján olcsóbb is lehetne: ő inkább üzletpolitikát sejt az ár mögött: egy potom árú alapgéphez drágább kiegészítő berendezéseket gyártanak. A gyártó szempontjából egyébként nagyon okos üzletpolitika.

Magyarországi ár – 2,7

2 KÍN

A Bizományi Áruház elfogadott árait vettük alapul (Interface 20 ezer, Microdrive 25 ezer forint) és – szokás szerint – inkvizítoraink elégedetlenek voltak. Érdekes a szórása az osztályzatoknak, hiszen a kettes és a négyes között minden érdemjegy megtalálható. Miután a vérmesebbek számon kérték, hogy ki merészelt négyest ad a magyarországi árra – kiderült, hogy a Sinclair típusú gépeknél lehet-





ZX MICRODRIVE

séges kétféle megközelítés most is bejött. Akik egyszerűen a nyugati árakhoz viszonyítanak, felháborítónak tartják a magas átszámítási kulcsot. Akik azonban abból indulnak ki, hogy még így is ez Magyarországon a legolcsóbb – majdhogynem elérhető árú – gép, magasabb osztályzatot adtak. Így érthetőbb a két inkvizítor által adott négyes.

Kezelhetőség – 3,7

3 KÍN

Ebben akínban aztpróbáltuk vizsgálni, hogy a teljes rendszer (számítógép – interface – háttérmemória) milyen egyszerűen használható, milyen gyorsan és könnyen érhető el programból vagy direkt utasítással. Mint az osztályzatból kitűnik, inkvizítoraink nem voltak elégedettek ezzel, volt aki így fogalmazott: „sok hókuszpókusz kell ahhoz, hogy egyáltalán megmozduljon"! A tény igaz, jó néhány csillagot és idézőjelet kell beírni, mire a Microdrive megtalál valamit, mégis van magyarázata a jobb osztályzatoknak is. Ezek az újonnan kifejlesztett eszközök egy meglehetősen bonyolult rendszer kiépítését is lehetővé teszik. Az Interface 1 konstrukciója olyan, hogy összesen 64 Spectrum számítógép összekapcsolására alkalmas, egy alapgéphez pedig maximálisan nyolc Microdrive köthető. Így érthetően bonyolultabb utasításokra van szükség, hogy a több lehetséges csatlakozás közül a gép mindig választani tudjon. Kétségtelen, elegánsabb megoldás lett volna, ha egyszerűbb rendszer esetén egyszerűbb utasítással mozgatható a gép, de talán elnézhető ez a kis testi hiba. Érdekes adat, hogy Angliában éppen azért fogadták el iskolai számítógépnek a Spectrumot, mert ezt a bizonyos 64-es kiépítést lehetővé teszi. Így ugyanis – egy képernyőre dolgozva a tanár és a diákok közösen írhatnak programot, oktatási célokra tehát kiválóan megfelel. Örömteli hír lenne, ha ugyanezt a hazai iskolaszámítógépről is elmondhatnánk. Reménykedve néztek egymásra inkvizítoraink, amikor azt kérdeztük, van-e valakinek tapasztalata ilyen összekapcsolásról, sajnos azonban komolyabb szervező munkára lenne szükség ahhoz, hogy itthon ilyen mennyiségű gépet egyszer összeszedjenek. Talán majd egy másik vallatásnál...

Kompatiblitás – 4,3

4 KÍN

Megközelítőleg hasonló kín sem szerepelt eddig soha Vallatónkban, aminek alapvető oka az, hogy a személyi számítógépek egymással egyáltalán nem kompatibilisek, tehát amilyen program az egyik gépen futtatható, azt a másikra át kell írni. Ez gyakran még az azonos típuscsaládba tartozó berendezéseknél is így van (ZX 81, Spectrum vagy VIC 20, Commodore 64). Az azonban jogos igény, hogy egy alapgép többféle háttérmemóriából azonos programokat be tudjon olvasni. Sajnos a Spectrum Microdrive konstruálásánál ezt az igényt nem tudták kielégíteni. A régebben készült, normál kazettán tárolt programok nem olvashatók be a gépbe, csak akkor, ha az Interface 1-et és a Microdrive-ot lekapcsoljuk. Így persze az átjátszás is meglehetősen bonyolult, programozói varázslatokat kell végezni ahhoz, hogy egy kazettán tárolt régebbi program Microdrive-ról futtatható legyen. Ennek ellenére a kompatibilitás magas osztályzatot kapott, ami talán annak az eredménye, hogy inkvizítoraink hozzáértő szakemberek, így a „programlopás" és -másolás összes fortélyát jól ismerik. Az egyszerű felhasználó azonban sokat szenvedhet ezzel.

A kazetta konstrukciója – 4,1

5 KÍN

Sok találgatás előzte meg a Microdrive megjelenését. Voltak, akik biztosan tudták, hogy lemezt rejt az igazán „mikroszkopikus" méretű műanyag doboz. Mások speciális szalagot tippeltek. Ma már nem titok, hogy egy kitűnő minőségű szalagról van szó, de a kazetta konstrukcióját inkvizítoraink közül is csak egy ember ismerte. Ennek oka talán az, hogy a 2000 forintos kazettát szórakozásból nem szokás szétszedni. Szalagszakadás miatt került sor a tanulságos operációra, amely során kiderült, hogy kb. 5 méter hosszú végtelenített szalgot tartalmaz a kazetta. A végtelenítés rendszere nem különbözik a kisméretű, gyerekeknek készült filmnéző berendezésektől vagy – akik ezt jobban ismerik – az AKAI cég által bevezetett nyolcsávos Cartridge elnevezésű magnetofonkazettától. A szalag a fej előtt elfutva egy nagyobb átmérőjű dob külső palástján folyamatosan tekeredik, majd a dob belsejében alul, egyszer megtekerve kerül vissza a fejhez. Mindebből tehát kiderült, hogy a konstrukció nem új, nem különösen szellemes, ami mégis indokolja a magas osztályzatot, az inkább a méret. Ha néhány évvel ezelőtt egy számítástechnikusnak azt mondták volna, hogy ötször öt centiméterben 100 kbyte elfér, nyilván kissé multságosnak találta volna a fantazmagóriát. A tény viszont tény, benne van. A kazetták között némi eltérés tapasztalható, amit inkvizítoraink többféleképpen magyaráznak, az egyik lehetséges indok az, hogy a szalaghossz nem pontosan egyforma minden kazettában. Néhány centiméteres eltérés már kbyte-okat eredményezhet. Két megjegyzés inkvizítorainktól a konstrukcióra vonatkozóan „zseniálisan egyszerű!" és „primitíven megbízhatatlan". Műszaki érdekesség, ami a kazettához kapcsolódik, de inkább a Microdrive intelligens voltát bizonyítja, hogy amennyiben véletlenül meggyűrődik a szalag valahol, arra a területre a drive automatikusan nem rögzít információt. Így esetleg egy-két kbyte-tal csökken a kapacitás, de biztosan jó a felvétel.

A kazetta ára a nyugati piacon – 3,2

6 KÍN

(5 angol font volt a vallatás idején. Azóta csökkent.) Az osztályzat nem valami fényes, ami azt jelenti, hogy a kazetta még kint is elég drága. Inkvizítoraink megegyeztek abban, hogy a konstrukció és a benne levő anyag nem indokolja a magas árat. Valószínűbb, hogy itt ügyes üzletpolitikáról van szó: olcsón vásárolhat bárki alapgépet, va perifériák már kicsit drágábbak, amiből pedig a legtöbb kell, tehát a kazetta ára indokolatlanul magas. Ez azért is bosszantó, mert természetesen ezt a típusú kazettát kizárólag a Sinclair cég készíti.

Kazetta magyarországi ára – 1,5

7 KÍN

(2000 Ft) Az osztályzathoz sok kommentár nem kell, úgy látszik, a fentiekben leírt monopolista üzletpolitika itthon még eredményesebb. Bár valaki úgy fogalmazott, hogy „ez csak félreértés

NAGYLUCSKAY LÁSZLÓ: Szeretném, ha minden rajzasztalra kerülhetne egy!

ILA LÁSZLÓ: Régóta várt eszköz, kellemes vele dolgozni!

BERTOLF JÓZSEF: Szervizelése sajnos megoldhatatlan!

MIHÁLYFI JÁNOS: A Commodore-nál 10–15-ször gyorsabb!

ERDEI LÁSZLÓ: Szeretném, ha a Sinclair kicsit strapabíróbb kivitelben is gyártaná a gépet. Mert aki komolyabb munkára akarja használni, az csak akkor tegye, ha forrón szereti a gépet!

RAUSCHER ATTILA: Többet reméltem, és többet is ígértek!

LIPOVECZ IVÁN: Szerintem összemérhető a komolyabb rendszerekkel is!

KUBISCH FERENC: Csoda, de jól működik!



A SPECTRUM MICRODRIVE
VALLATASANAK EREDMENYE

KINOK	1 SZEKELY JENO FOISK. TANAR	2 ILA LASZLO VILLAMOSMERNOK	3 ERDEI LASZLO MUSZAKI VEZETO	4 NAGYLUCSKAI LASZLO UZEMMERNOK	5 KUBISCH FERENC GM. TAG	6 LIPOVECZ IVAN GM TAG	7 BARTOLF JOZSEF TERVEZO MERNOK	8 RAUSCHER ATTILA FEJLESZTO MERNOK	9 GERO GABOR SZAM.TECHN. M.TARS	10 MIHALYFI JANOS MERNOK-TANAR	ATLAG
1 .KIN:AR - NYUGATON	4	3	5	5	4	4	4	4	4	4	4.1
2 .KIN:AR - ITTHON	3	1	2	4	4	2	2/3	3	-	3	2.7
3 .KIN:KEZELHETOSEG	3	3	4	4	4	4	4	3	-	4	3.7
4 .KIN:KOMPATIBILITAS	4	-	5	5	5	5	3	2	-	5	4.3
5 .KIN:KAZETTA KONSTRUKCIO	4	4	5	4	4	4	3	5	3	5	4.1
6 .KIN:KAZETTA ARA NYUGATON	4	3	4	4	2	3	4	2	2	4	3.2
7 .KIN:KAZETTA ARA ITTHON	2	1	1	1	2	2	1	1	2	2	1.5
8 .KIN:RENDSZER MEGBIZH.	4	3	4	4	5	4	3	3	3	4	3.7
9 .KIN:DOKUMENTACIO	4	2	5	5	4	5	5	5	1	5	4.1
10 .KIN:GYORSASAG	4	5	5	4	5	4	5	3	4	4	4.3
11 .KIN:SZOLGALTATASOK	4	-	5	4	4	4/5	4	2	-	5	4.1
+ 1 KIN:SZUBJEKTIV VELEMENY	4	4	5	5	5	5	3	3	-	4	4.2
ATLAG	3.7	2.9	4.2	4.1	4.0	3.9	3.5	3.0	2.7	4.1	3.6

A rendszer megbízhatósága – 3,7

8 KÍN

Ez ismét olyan téma, amit önállóan nem lehet vizsgálni, csak a hármas rendszert együtt. Alapvető probléma, amit többen felvetettek, hogy a felszerelt Interface 1 az alapgéppel együtt erősen melegszik. Hallottunk erre már zseniális műszaki megoldásokat (gyufásskatulyát kell alátenni, modernebbek radírgumit is használhatnak), de eddig még a Sinclair cég nem vásárolta meg ezeket a szabadalmakat. Másik gyakori hibalehetőség a csatlakozópontoknál adódik, ráadásul, ha egy csatlakozó eltörik, nem lehet újat 200 kilométeren belül beszerezni.

A géphez adott dokumentáció – 4,1

9 KÍN

A gépkönyv kizárólag használati leírást tartalmaz. Inkvizítoraink véleménye szerint ez ahhoz éppen elég, hogy használni lehessen a berendezést, átlagfelhasználónak tehát megfelel. Aki komolyabban szeretne belepiszkálni a dolgokba, az viszont semmit nem tud meg, még alapvető műszaki adatok sincsenek feltüntetve. Hiába no, a Sinclair cég tud titkot tartani.

Gyorsaság – 4,3

10 KÍN

Itt az adatok elérhetőségi sebességéről van szó, ami nem azonos az adatátviteli sebességgel. Tehát arról, hogy egy megadott programot vagy más rögzített információt milyen gyorsan talál meg a Microdrive. Többen szemrehányóan említették, hogy a tervezés időszakában nagyobb sebességet ígért a cég, a valóságos viszont a 7 másodperces maximális elérési idő. Az osztályzat mégis elég magas, hiszen ez még így is gyorsabb, mint például a Commodore-rendszer.

Szolgáltatások – 4,1

11 KÍN

Azt igyekeztünk osztályozni, hogy a rendszer milyen lehetőségeket rejt magában, milyen szolgáltatásokat képes nyújtani. Inkvizítoraink elégedettek, hiszen ez a rendszer csírájában, néha nehézkesen, de lényegében mindent tud, ami a komolyabb gépek. Ugyanakkor van néhány specialitása is. A 64 Spectrum összekapcsolási lehetőségét már említettük, nem volt szó azonban még arról, hogy egy alapgéphez maximális nyolc Microdrive is köthető. A Sinclair cég ajánlása szerint három Microdrive-val végezhetők kényelmesen az adatfeldolgozási feladatok, egyik inkvizítorunk véleménye szerint azonban ez nem jelent 300 kbyte háttérmemóriát, csak háromszor 100-at. Egyébként a nyolc microdrive-os kiépítésről senkinek nincs tapasztalata, ennyit egy helyen még senki sem látott.

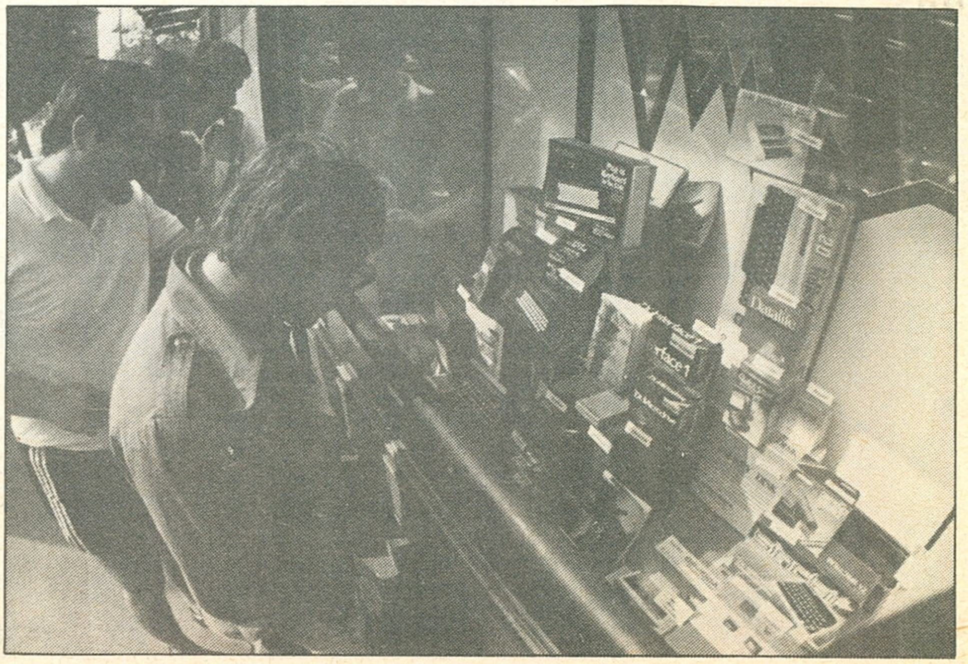
Szubjektív vélemény – 4,2

+1 KÍN

Kommentár nem kell hozzá, inkvizítoraink többsége megszerette ezt az első látásra mulatságos, tudására nézve azonban bámulatos masinát. Valamennyien egyetértettek abban, hogy a Spectrum zseniális konstrukciója mellé törvényszerű volt kifejleszteni a gyors háttérmemóriát, és abban, hogy ezzel a gép lehetőségei nagymértékben növekedtek. A Sinclair cég ismét olyat csinált, ami olcsó, de a néhány hónapos tapasztalat alapján tisztességesen működik.

HOZZÁSZÓLÁS

Kedves Inkvizítoraink!
A Sinclair Microdrive kinzásakor, úgy látom, néhány dologról megfeledkeztetek.
Helyes, hogy a Microdrive-ot az Interface 1-gyel együtt értékeltétek. Azonban: miért nem vizsgáltátok a hálózati lehetőségeket? Nem kellett volna hozzá 64 gép – már kettővel is ki lehetett volna próbálni. Az Interface 1-be beépített RS 232 C típusú interface-ről pedig szót sem ejtettek! Pedig az együttes árát csak úgy lehet értékelni, ha tudjuk: egy Spectrumhoz az RS 232 C interface önmagában is 20–30 fontba kerül. (Segítségével printert, modemet és még sok mást lehet a Spectrumhoz illeszteni.)
Nagyon hiányoltam, hogy a file-kezelésről egy árva szó sem esett. Valóban mindent ugyanúgy tud a Microdrive mint egy floppy – vagy jobban?! És végül: azt egyáltalán nem hiszem el, hogy a Microdrive-os gépbe ne lehetne kazettáról is programot beolvasni. Vagy



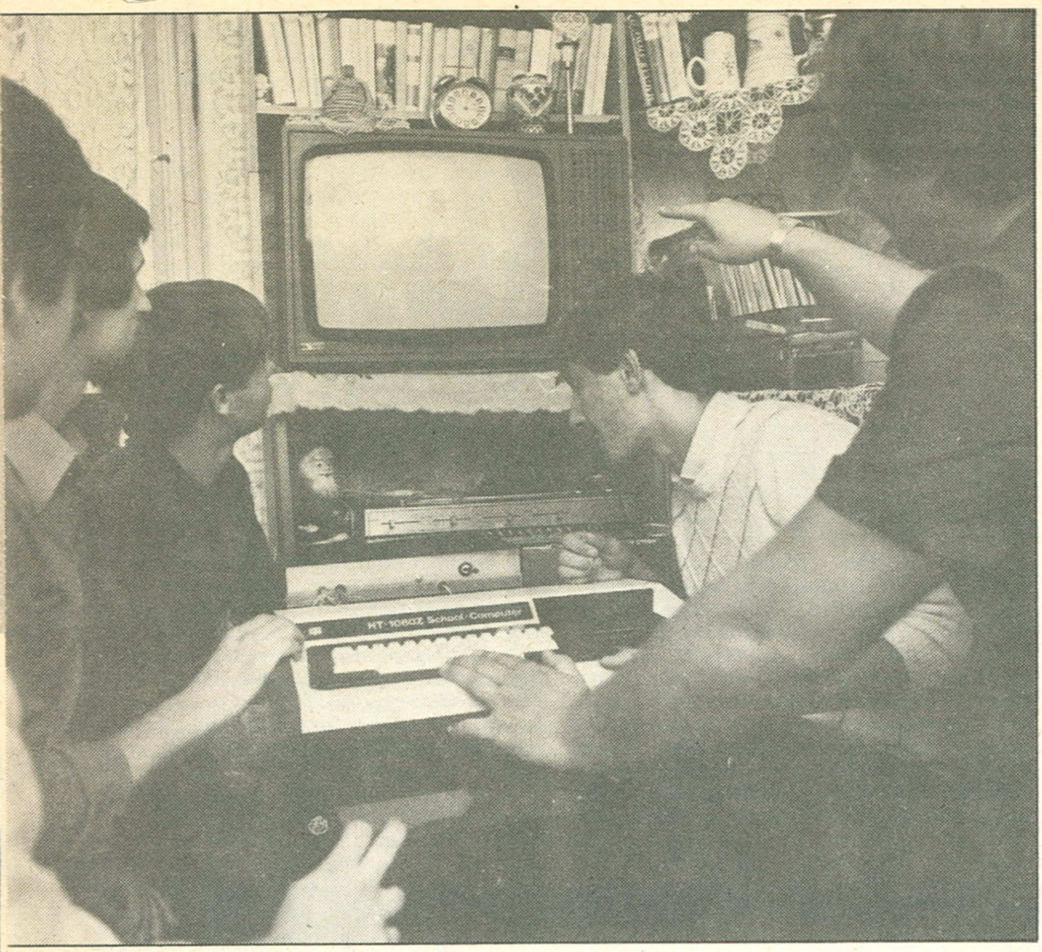
félreértettem volna a 4. kínban irottakat? Általában: jó, hogy új és még csak terjedő gépeket és perifériákat is kinoztak. De érdemes lenne ezt az alkalmat némi ismeretterjesztésre is felhasználni – esetleg a terjengős lelkizések rovására is.
Baráti üdvözlettel:
Szekfű András

Kazetta – Microdrive

Tisztelt Szerkesztőség!
Rovatukat mindig nagy érdeklődéssel olvasom, s miután több mint egy éve dolgozom ZX Spectrummal, két hónapja pedig a Microdrive-ot is használom, az utóbbi „Vallató"-jára különösen kiváncsi voltam. Vizsgálati szempontjaik – úgy hiszem – most is eléggé átfogóak, s ennek megfelelően véleményüket is árnyaltnak érzem, amely többé-kevésbé az én eddigi tapasztalataimmal is megegyezik, alapvető kérdést, a kazetta-Microdrive kompatibilitást kivéve. Önök szerint ui. a régebben kazettára írt programok csak akkor olvashatók be a gépbe, ha ezt megelőzőleg az Interface 1-et, ill. a Microdrive-ot lekapcsoljuk. Eltekintve attól, hogy a „lekapcsolás" szó feltételez egy „lekapcsolót", ilyen pedig nincs beépítve, de az alapgép és a vizsgált perifériák szétválasztása is meglehetősen körülményes, amennyiben az Interface 1 korrektül, az előírásoknak megfelelően van csatlakoztatva, ez az állítás nem felel meg a tapasztalataimnak. Mindegyik korábban írt programomat különösebb nehézség nélkül sikerült Microdrive kazettára átírnom, s semmilyen nehézséget nem tapasztaltam a Microdrive és a kazettás magnó egyidejű használatában. Tény, hogy a Microdrive mellett volt egy figyelmeztetés, hogy bizonyos programok kazettáról való beolvasásánál előfordulhatnak nehézségek, ha az Interface 1 csatlakoztatva van, de ezek elkerülhetők, amennyiben a LOAD utasítást egy NEW és ENTER utasítás előz meg. A saját kedvező tapasztalatomat két kiváló, forgalomban levő program Microdrive-változata is megerősít. A Masterfile-program egyidejű választási lehetőséget ad a kazettás magnóra vagy a Microdrive-ra történő adattárolásra, a Tasword Two szövegkezelő programot pedig a felhasználónak kell néhány BASIC sor megváltoztatásával Microdrive-ra alkalmassá tennie. Mindkettőt magnókazettán forgalmazzák. Hogyan lehetne egyáltalán betölteni ezeket magnóról, s utána Microdrive kazettára kiírni, ha közben az Önök instrukciójának megfelelően a gépet ki kellene kapcsolni (hogy az Interface 1-et, ill. a Microdrive-ot kiiktassuk, ill. ismét csatlakoztassuk)? Egy ponton valóban nem kompatibilisek a korábbi programok: amennyiben REM utasításban tárolt gépi kódú programsorokat tartalmaznak: ui. az Interface 1 és a Microdrive használata plusz 512 byte-ot foglal le az ún. gépi változók számára, tehát a BASIC programsorok gépi címzései ennyivel eltolódnak, a REM-ben tárolt utasításokat pedig az Interface címzései felülírják. De ez a gond sem a beolvasásnál jelentkezik, hanem a futtatásnál. Egyszerű BASIC programok esetében, vagy amennyiben a gépi kódú program a RAMTOP fölé lett címezve, kompatibilitási nehézség elvben nem fordulhat elő. Természetesen több részből álló programoknál, ha az újabb beolvasás(ok)ra maga a program ad utasítást, a magnóról történő beolvasás után a Microdrive-ra való kiíratás előtt a megfelelő utasításokat értelemszerűen át kell írni. De ez a szintaxis és nem a kompatibilitás kérdése.
Tapasztalataim alapján: úgy vélem tehát, hogy a kompatibilitásra vonatkozó megállapításuk túlzottan sommás, s ezért riasztólag hathat azokra, akiknek elég sok „átírnivaló" programjuk van. Bár kétségtelenül annyival gyorsabb Microdrive-val dolgozni, mint kazettás magnóval, hogy hosszú távon még az is időmegtakarítást eredményez, ha netán újra be kell „pötyögni" egy programot.

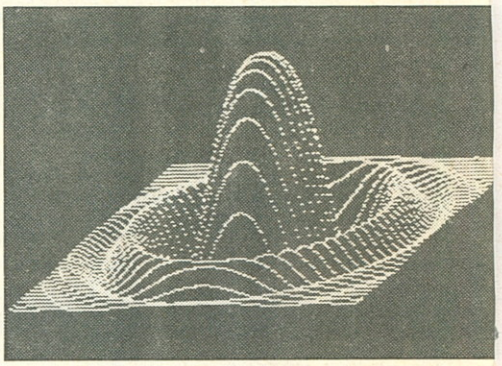
Jeney Zoltán

SZUPER VALLATÓ



kívánunk grafikát, akkor is érdemes és hasznos a készülék kapcsolása, ugyanis ez esetben 16 kbyte-os gépünk 48 kbyte-osra bővül. Azaz 32 K memóriabővítést tartalmaz a berendezés. Ezután még mindig újabb választási lehetőségünk van a grafikus használat esetén.
a) A 48 kbyte-ból 16-ot használunk a grafikára – s négy színnel dolgozhatunk, a maradék 32 kbyte pedig megmarad használható memóriaterületnek.
b) 32 kbyte-ot használunk a grafikára, ekkor már 16 színnel dolgozhatunk, s a maradék 16 kbyte használható memóriaterületünk. (A használandó színek számuktól függetlenül 256 árnyalat közül választhatók ki!)
Használati élmények: a berendezés úgy önmagában természetesen assemblerben programozható. Ám hogy erre ne legyen szükség, Heiczman Viktor elkészítette azt a gépi kódú programcsomagocskát, amelynek a beolvasása után meglehetősen egyszerűen programozható grafikánk BASIC-ből. Hogy ennek mi a módja, s mennyire egyszerű, azt jól szemlélteti írásunk későbbi részében a bemu-

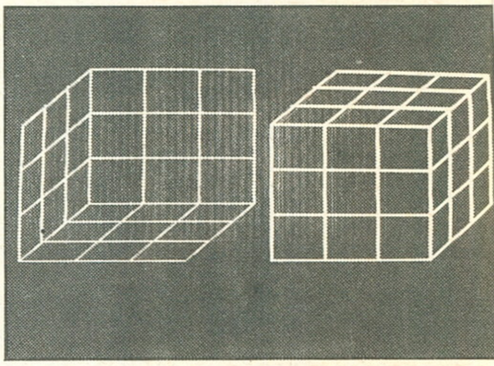
1. sz. kép



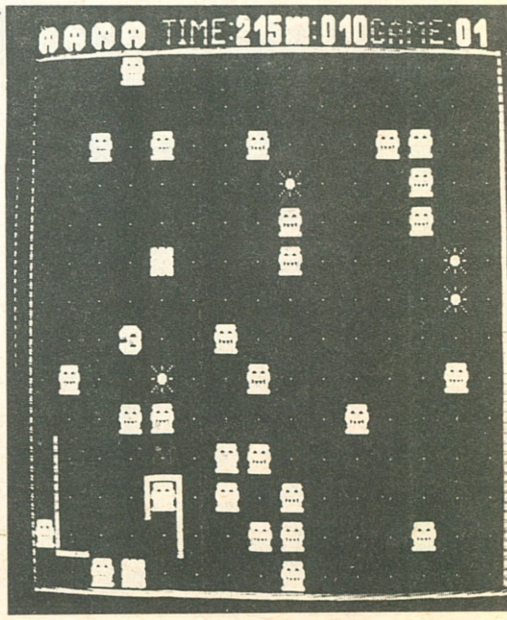
BIT-LET-ünkben egyszer egy gazdasági munkaközösségnek a ZX 81-hez gyártott jelfrissítő készülékét vizsgálgatták szerkesztőségünk szakértői, munkatársai. Utána több hasonló vizsgálatra invitáltak bennünket új hardverkiegészítéseket gyártó munkaközösségek, fejlesztők. Igazság szerint nem zárkózunk el az ilyenfajta vallatásoktól, így hát szívesen fogadunk efféle invitálást, megkeresést. Annak jogát azonban fönntartjuk magunknak, hogy az ajánlatok közül melyiket fogadjuk el. Alapelvként azonban leszögezhetjük, hogy döntésünket ilyenkor nem valamiféle szubjektív szempontok vezérlik, hanem annak mérlegelése, hogy az adott hardver mennyire tarthat számot közérdeklődésre, milyen széles felhasználói réteget érint, s hogy a probléma, amelyet a berendezés, kiegészítés megold, valóban probléma-e? Monori István villamosmérnök azzal keresett meg bennünket, hogy készítettek egy hardvert, amely a HT 1080Z iskolaszámítógéphez készült, s amely nagy felbontású, színes grafikát képes produkálni. Kölcsönkaputnk egyet e MICOLOR 01 névre hallgató berendezésből, és megvizsgáltuk.
A vizsgálatra összehívott kis stábunkban volt a Tudományszervezési és Informatikai Intézet munkatársa, Mihályfi János – egyben az alábbi „értékelés" megfogalmazója – egy egyetemi hallgató, aki BIT-LET-ünk posta rovatának vezetője – Halász Péter –, és hogy az igazi felhasználók is képviselve legyenek, két középiskolás diák: Egri Tibor és Szabó Attila, a Kossuth Lajos Gimnáziumból. Jelen volt a vizsgálódáson a készülék készítője, Monori István is, s a kiegészítő szoftvert író Heiczman Viktor.

Az összeszerelés különösebb gond nélkül zajlott. Egyszerűen a bus csatlakozón keresztül kapcsolható össze a MICOLOR és a HT. Természetesen a HT monitor kivezetését ezúttal nem a tévéhez kell csatlakoztatni, hanem a kiegészítő hardverhez, s ennek a kivezetését kell azután a televízióhoz, illetve annak monitorbemenetéhez kapcsolni. Némi szomorúsággal vettük tudomásul, hogy a berendezés képszíneinek vizsgálatáról ezúttal le kell mondanunk, mert a masinából kijövő jel csak az RGB típusú bemenethez illeszkedik megfelelően, s a mi tévénknek csak videóbemenete volt. (Ez a tévében történő egyszerű átalakítással megoldható lett volna, de ezt az átalakítást ezúttal nem kívántuk elvégezni.)
A konstruktőr megnyugtatott bennünket, hogy egyrészt az átalakítás nem komoly, nem is költséges, s a tévé egyéb funkcióit sem veszélyezteti, mert az átalakított gép kap egy egyszerű kapcsolót, amellyel kiválaszthatjuk a megfelelő funkciót: normál állás vagy a berendezés kívánta átalakított változat. Másrészt afelől is megnyugtattak bennünket, hogy a berendezés színei megfelelnek a mai színestévé-technika s az ipar színestelevízió-gyártási színvonala adta lehetőségeknek. (Tapasztalatunk szerint a mikroszámítógépek igazán jó színlehetőségeinek kibontakozását, kihasználását sokkal inkább az utóbbi gátolja, mintsem a számítógép tudása. – A szerk.)
A berendezés összerakása közben egy pillantást vetettünk annak külső alakjára is. Ez a szemlélés különösebb eredményt nem hozott, mivel a készülék eme első darabjának kivitele udvariasan fogalmazva egyszerű, kevésbé udvariasan: kivitel még nincs, csak doboz.
1. sz. kellemes meglepetés: az összeszerelés, szemlélődés és sajnálkozás – ugyanis a színek miatt – utána az első kellemes meglepetés következett. Megtudtuk ugyanis, hogy a készüléknek több funkciója van. Ha nem

2. sz. kép

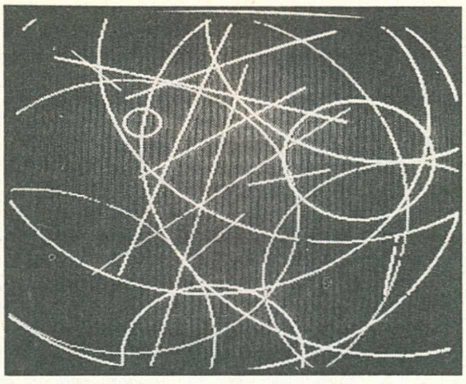


3. sz. kép

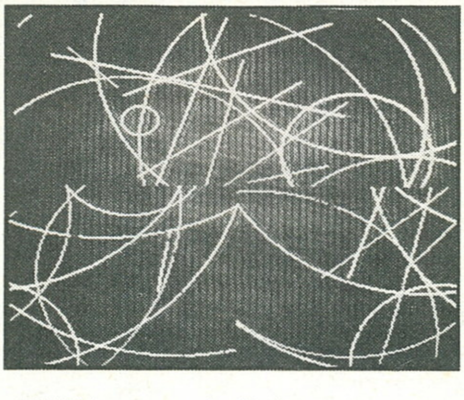


MICOLOR 01





4. sz. kép



5. sz. kép

tatott példaprogram és a hozzá tartozó ábra. Amit még tudni kell az érdeklődőknek, potenciális felhasználóknak, hogy az alfanumerikus és a grafikus képernyő váltása egy egyszerű kapcsolóval történik. A kettő keverésére tehát nincs lehetőség.

A használat milyenségét elsősorban diákvendégeink tapasztalatai alapján minősítjük. Nos, ők kb. egy óra ismerkedés után már teljesen otthonosan dolgoztak a géppel, a legkülönbözőbb formákat rajzolgatták, s a legképtelenebb ötletekkel álltak elő, amelyeket – ismerve őket – bizonyára meg is valósítottak volna, ha van rá néhány éjjelük és nappaluk. (De hát a berendezést csak kölcsön kaptuk, s ráadásul aludni is kell néha – még egy középiskolásnak is.)

Lényeges „apróságok": mind ez ideig „titkoltuk", most már eláruljuk, mennyi is a felbontása a berendezésnek: 256x256 képpont. Hogy ez sok vagy kevés, ki-ki döntse el. Szerintünk épp elegendő ahhoz, hogy dolgozni lehessen vele.

Ugyancsak lényeges dolog, hogy az úgynevezett képsíkok száma választás szerint egy vagy kettő. E képsíkok külön-külön és együtt is programozhatók. A képsíkok egyike tetszés szerinti irányba mozgatható. (Ráadásul a mozgatható képsík négy részre osztható, s e részek külön-külön is mozgathatók!) És végül két „apróság" a szakembereknek: 1. A sor és a kép visszafutása két bit segítségével regisztrálható. 2. A képmemóriába való írás a képfrissítést nem zavarja – azaz a kép nem villog.

Két kevésbé lényeges apróság: A készüléknek beépített tápegysége van; a készülék méretei: 35x25x6 cm.

A közölt képekről és a programozás módjáról:

Az 1. sz. képen egy rövid BASIC program eredménye látható. Az egyenes és görbe vonalakat egyaránt tartalmazó ábra jól szemlélteti a felbontás milyenségét, az ábra plasztikusságát.

A 2. sz. kép a képsíkok mozgását hivatott illusztrálni. Az itt látható két vetület ugyanis összecsúsztatható, s ez esetben a kocka teljes térhálója látható. A 3. sz. fotón egy kész játékprogram látható. Ez sok mindent bizonyít! És végül az utolsó két fotóról és a hozzájuk tartozó programlistáról: ezt a programocskát amolyan demonstrációs céllal írtuk elsősorban a két jelenlévő diák közreműködésével. Természetesen a program csak az említett gépi kódú segédprogram használatával működik. Amint látható, ez a segédprogram a HT1080Z BASIC-jének disckezelő utasításait használja grafikus utasításként. Ez pillanatnyilag jó megoldás, hiszen drive-illesztés legfeljebb fehér hollóként fordul elő az iskolaszámítógéphez. (Ezzel együtt, a jövőre is gondolva, jobb megoldást is el tudtunk volna képzelni.) Az utasítások jelentése a következő:

– PUT (képernyő törlése)
– GET (pont rajzolása)
– OPEN (egyenes rajzolása)
– FIELD (kör rajzolása)

A szükséges paramétereket POKE utasítással kell a memóriába elhelyezni.

Ez a program véletlenszerű egyeneseket és köröket rajzol és mozgatja a képsíkot. Sorba menve: a 7-es sorban történik a színregiszterek feltöltése, a szükséges színkódokat a 130-as sorból olvassa ki programunk. A 8-as sorban a mozgatható képsíkot alapállapotba helyezzük. A 9-es sorban töröljük a két képsíkot, s a POKE 37761,0 vagy a 37761,255 jelöli, hogy melyik képsíkot lehet törölni. A 10-es sorban történik véletlenszerűen az egyenes, a kör rajzolása vagy a képsíkcsúsztatás funkcióra való ugrás. A 20–24-es sorban rajzoltatjuk meg az egyeneseket. A program véletlenszerűen kiválaszt két végpontot (X1, Y1 és X2, Y2) és egy színt (SZ), majd ezek értékének megfelelően megrajzolja az egyenest az OPEN utasítás hatására.

A 30–35-ös sorokban a körrajzolás történik az előzőhöz hasonló módszerrel. A 100–120 sorok a képsík mozgatását végzik. Az OUT 12,9128–I a bal, illetve jobb félkép függőleges eltolásának mértékét adja meg. Az OUT 130,I és OUT 131,64–I pedig a felső, illetve az alsó félkép vízszintes mozgatását végzi. A két fotó, amely a program futtatása során készült, illusztrálja a képsíkok elmozdulását is, hiszen a két kép elkészülése közben újabb grafikai elemek nem kerültek a képernyőre, csak éppen elmozdítottuk vízszintes irányba az egyik képsíkot. (Ha megfigyelik, jól látható a kép közepén az a vízszintes „vonal" amelynek mentén az elmozdulás megtörtént.)

Úgy gondoljuk, egyelőre ennyit épp elég tudni olvasóinknak a MICOLOR 01 használatáról. Már így is van némi lelkiismeretfurdalásunk. Lehet ugyanis, hogy túlságosan felcsigáztuk diákok és tanárok fantáziáját, s ugyanakkor egyelőre nem nagy a valószínűsége, hogy túl sok helyen lehetne ott a közeljövőben ez a berendezés.

S mert minden vallatónk végén szokott lenni úgynevezett szubjektív vélemény, engedtessék meg, hogy ezúttal is legyen legalább annyi, hogy a MICOLOR 01 egyéb erényei, könnyű és egyszerű használhatósága mellett legfőbb erénye, hogy elkészült. Hogy mire képes – jó lenne, ha erre a gyakorlat adna választ, maguk a diákok és a tanárok lehetnének a legfőbb tesztelők – mielőbb.

Fenti vallatót hónapokkal ezelőtt közöltük először a BIT-LET-ben. Azóta történt egy és más a MICOLOR háza táján. Először is eladtak belőle kb. 30 db-ot. Másodszor is a gyártók fölkészültek a 64 kbyte-os HT megjelenésére. Harmadszor is – s ez külön öröm – a gyártók megfogadtak néhány, a vallatáson elhangzott észrevételt. Monori István kérésünkre néhány sorban összefoglalta a legfontosabb MICOLOR színű híreket.

1. A MICOLOR 01 működik a 64 kbyte-os HT géppel is: A régi HT-vel 64 kbyte-ra, az új HT-val 96 kbyte-ra nő a memóriaterület.

2. Az eredeti és a grafikus kép programból, vagy a két SHIFT billentyű egyidejű lenyomásával kapcsolható át, tehát megszűnt a vallatáson kifogásolt mechanikus kapcsoló.

3. Elkészült egy grafikus programcsomag, amely a normál BASIC parancsokhoz hasonlóan egyszerű rajzolási lehetőségeket teremt. Szolgáltatásai a C 64-es gépet támogató programokhoz hasonló.

– Pont, kör, egyenes rajzolása;
– Szövegmegjelenítés, nagyítás X, Y irányban, dőlésszög beállítása;
– Változtatható karaktergenerátor, ékezetes betűk, cirill betűk (pl. orosz–magyar szótár készítése);
– Négyféle területfeltöltő utasítás.

4. Külön kimenetet készítettünk ORION, illetve VIDEOTON színes televíziókhoz.

5. Kisebb lett a méret és esztétikusabb a kivitel.

Záradék: mindennemű reklámetikai megfontolás mellett, alatt és fölött szerkesztőségünk önkímélés és közvetítés helyett inkább e helyütt közli, hogy a berendezés megrendelhető Monori Istvánnál – Budapest 1165 Géza u. 15/B. (Ára: 15 000 forint)

```
10 FOR I=160 TO 175 : READ W : OUT I,W : NEXT
20 FOR I=128 TO 131 : OUT I,0 : NEXT
30 POKE 32761,0 : PUT : POKE 32761,255 : PUT
40 Q=RND(50) : IF Q<3 THEN 120 ELSE A=INT(RND(2)) : ON A GOTO 50,80
50 X1=RND(255) : X2=RND(255) : Y1=RND(255) : Y2=RND(255)
60 POKE 32742,X1 : POKE 32740,Y1 : POKE 32746,X2 : POKE 32744,Y2 : OPEN
70 GOTO 40
80 X=RND(255) : Y=RND(255) : RA=RND(128) : SZ=4*INT(RND(3))
90 POKE 32740,SZ
100 POKE 32726,X : POKE 32724,Y : POKE 32728,RA : FIELD
110 GOTO 40
120 FOR I=0 TO 128 : OUT 128,I : OUT 129,128-I : FOR W=0 TO 30 : NEXT W,I
130 FOR I=0 TO 64 : OUT 130,I : OUT 131,64-I : FOR W=0 TO 30 : NEXT W,I
140 GOTO 20
150 DATA 0,0,0,0,35,35,35,35,110,110,110,110,255,255,255,255
```

benchmark

Számítógépes újságokban gyakran olvashatunk titokzatos „BENCHMARK" programokról, amelyeknek általában csak a futási idejét közlik. A magyar olvasók, ha hozzá is jutnak ezekhez a lapokhoz, legtöbbször értetlenül állnak a közölt időadatok előtt, nem tudván mire véln iazokat.

Mi is az a „BENCHMARK"? Evvel az összefoglaló névvel egy nyolc BASIC programból összeállított sorozatot értenek, amelyek különböző számítógépek és BASIC-interpreterek (fordítóprogramok) sebességének összehasonlítására szolgálnak. Maga a kifejezés két angol szó összevetéséből származik (bench-munkapad, mark-jel), és nem is valami számítógépes „futóverseny" lebonyolítása, hanem csak a relatív sebességek becslése.

Ezeket a programokat már 1977–78-ban összeállították, és azóta használják őket. Voltak közben kísérletek más időmérő programok bevezetésére is, de a legelterjedtebb ez a nyolc BENCHMARK program lett.

A sorozat úgy van felépítve, hogy minden program az őt megelőzőtől csak egy utasításban különbözik. Kivétel ezalól a nyolcas, amelyik a leggyakoribb transzcendens függvények vizsgálatára szolgál, és teljesen különáll a többitől. A másik hét program segítségével néhány jellegzetes BASIC utasítás végrehajtási idejét mérhetjük. Ugyanis, ha két egymás utáni, program futási idejét kivonjuk egymásból, megkapjuk annak az utasításnak a végrehajtási idejét, amelyben a kettő különbözik. A könnyebb mérhetőség érdekében a gép minden utasítást ezerszer hajt végre. Például ha a 3. futási idejéből kivonjuk a 2.-ét, megkapjuk, hogy mennyi idő alatt csinál a gép 1000 olyan értékadó utasítást, mint amilyen a 3. program 140 sorában látható. Természetesen ezekkel az adatokkal óvatosan kell bánni. A BENCHMARK programok csak a leggyakoribb BASIC utasításokat vizsgálják, azokat amelyek minden gépen megtalálhatók. Ezek alapján nem lehet összehasonlítani pl. a grafikai utasításokat, pedig ezek a személyi számítógépes alkalmazások nagy részénél meghatározó szerepet játszanak. Gondoljunk például a játékprogramokra, amelyekbe a mozgó ábrák lehelnek lelket! Az ilyen „extra" utasításokkal foglalkozó tesztprogramok megjelenésére még remény sincs, mivel a különböző gépeken akkora eltérések vannak, hogy eleve lehetetlen az összehasonlítás. Tehát a BENCHMARK időknek csak addig van jelentőségük, míg a programok nem használnak túl sok, az adott gépre korlátozott utasítást.

Más probléma is van! A BENCHMARK sebességek nemcsak a gép elektronikájától, hanem a BASIC interpretertől is függnek. Olyan gépeknél, ahol lehetőség van a BASIC interpreter kicserélésére egy jobban megírt interpreterrel esetleg sokkal jobb eredményt érhetünk el. Másik lehetőség a BASIC compilerek használata. Ezek olyan fordítóprogramok, amelyek a bonyolultabb nyelveknél és nagyobb számítógépeknél megszokott módon egyszerre fordítják le az egész BASIC-programot gépi nyelvre. Az így keletkezett gépi kódú programot futtatják azután. Ez természetesen sokkal gyorsabb az interpreteres változatnál, ahol a gép futás közben soronként bíbelődik a fordítással. Ilyen compiler még a ZX 81-hez is kapható! Akár nagyságrendekkel is javulhat a gép sebessége.

Tehát semmiképpen sem érdemes a BENCHMARK idők alapján választani számítógépet, még ott sem ahol erre lehetőség van! Azonban, ha figyelembe vesszük ezeket az időket, akkor a program kritikus lassú részeinek a gondos átírásával értékes percekkel esetleg napokkal csökkenthetjük az idegtépő várakozás perceit.

Végül a PERSONAL COMPUTER WORLD táblázata és saját méréseink alapján közöljük néhány itthon is ismert személyi számítógép BECNHMARK idősit.

Zoletnik Sándor

BM 1

```
100 REM Benchmark 1
110 PRINT "S"
120 FOR K=1 TO 1000
130 NEXT K
140 PRINT "E"
150 END
```

BM 2

```
100 REM Benchmark 2
110 PRINT "S"
120 K=0
130 K=K+1
140 IF K<1000 THEN 130
150 PRINT "E"
160 END
```

BM 3

```
100 REM Benchmark 3
110 PRINT "S"
120 K=0
130 K=K+1
140 A=K/K*K+K-K
150 IF K<1000 THEN 130
160 PRINT "E"
170 END
```

BM 4

```
100 REM Benchmark 4
110 PRINT "S"
120 K=0
130 K=K+1
140 A=K/2*3+4-5
150 IF K<1000 THEN 130
160 PRINT "E"
170 END
```

BM 5

```
100 REM Benchmark 5
110 PRINT "S"
120 K=0
130 K=K+1
140 A=K/2*3+4-5
150 GOSUB 190
160 IF K<1000 THEN 130
170 PRINT "E"
180 STOP
190 RETURN
200 END
```

BM 6

```
100 REM Benchmark 6
110 PRINT "S"
120 K=0
130 DIM M(5)
140 K=K+1
150 A=K/2*3+4-5
160 GOSUB 220
170 FOR L=1 TO 5
180 NEXT L
190 IF K<1000 THEN 140
200 PRINT "E"
210 STOP
220 RETURN
230 END
```

BM 7

```
100 REM Benchmark 7
110 PRINT "S"
120 K=0
130 DIM M(5)
140 K=K+1
150 A=K/2*3+4-5
160 GOSUB 230
170 FOR L=1 TO 5
180 M(L)=A
190 NEXT L
200 IF K<1000 THEN 140
210 PRINT "E"
220 STOP
230 RETURN
240 END
```

BM 8

```
100 REM Benchmark 8
110 PRINT "S"
120 K=0
130 K=K+1
140 A=K^2
150 B=LOG(K)
160 C=SIN(K)
170 IF K<1000 THEN 130
180 PRINT "E"
190 END
```

GÉPTÍPUS	BM1	BM2	BM3	BM4	BM5	BM6	BM7	BM8
ABC 80	1.1	2.3	11.1	12.1	12.6	17.7	23.9	136.0
APPLE II	1.3	8.5	16.0	17.8	19.1	28.6	44.8	107.0
APPLE III	1.7	7.2	13.5	14.5	16.0	27.0	42.5	75.0
BBC micro	1.0	3.1	8.2	8.7	9.1	13.9	21.4	51.0
COMMODORE 20	1.4	8.3	15.5	17.1	18.3	27.2	42.7	99.0
COMMODORE 64	1.6	9.7	18.3	20.3	21.8	31.5	49.5	115.9
COMMODORE CBM 8032	1.7	10.0	18.4	20.3	21.9	32.4	51.0	119.0
IBM Personal Computer	1.5	5.2	12.1	12.6	13.6	23.5	37.4	35.0
Hewlett Packard HP 85	1.8	3.8	16.3	16.5	17.7	30.0	44.8	127.0
Hewlett Packard HP 125	1.7	5.0	12.5	12.5	14.0	26.0	40.0	60.0
Tandy TRS–80 Color Comp.	2.0	11.3	22.2	23.9	27.0	41.5	61.1	130.0
Tandy TRS–80 Modell II	1.0	5.0	13.0	13.0	14.0	23.0	35.0	60.0
TRS–80 Modell I. Level II	2.7	11.6	28.0	28.5	31.3	51.9	81.0	117.0
ZX–81 (FAST üzemmód)	4.5	6.9	16.4	15.8	18.6	49.7	68.5	229.0
Sinclair Spectrum	4.8	8.7	21.1	20.4	24.0	55.3	80.7	253.0
Sharp MZ80B	0.6	4.5	8.5	11.5	13.0	19.0	27.5	50.0
Sharp MZ80K	1.4	9.4	16.3	22.5	25.4	36.8	51.1	102.0
Sharp MZ80A	1.5	9.2	16.4	22.8	25.6	37.7	55.0	101.0
Sharp PC3201	4.0	13.5	35.5	35.5	38.5	67.0	108.0	250.0
Casio fx9000	2.5	9.0	24.0	24.0	26.0	42.0	60.0	365.0
Atari 400/800	2.3	7.4	19.9	23.2	26.8	40.7	61.5	431.0
Texas TI 99/4A (alap)	3.0	9.0	24.0	24.8	26.2	61.9	84.6	384.0
Texas TI 99/4A (bővített)	6.5	18.5	40.0	40.1	42.0	98.4	140.3	240.0
Aircomp 16 (lassú)	2.0	10.0	19.5	21.0	23.0	26.5	57.0	75.0
Aircomp 16 (gyors üzem)	0.7	4.0	8.0	8.0	9.5	15.0	23.0	30.0
HT 1080Z iskola-számítógép	3.0	11.5	27.0	28.0	32.0	52.0	80.0	120.0
M08X	1.9	8.2	21.8	22.4	23.7	39.8	64.7	114.8
Proper 8	3.0	8.2	21.8	22.4	23.7	42.1	68.0	124.0
Proper 16	1.9	5.0	12.0	12.2	13.1	22.0	35.4	33.3
SIMON '68	0.92	2.88	8.51	8.61	10.8	13.3	21.2	41.3
PRIMO	1.6	7.7	18.5	19.2	21.1	33.6	51.6	82.9
Sinclair QL	1.97	5.36	9.46	9.27	11.92	24.06	42.58	21,03
Commodore 16	2.00	9.91	18.35	19.11	21.42	35.02	55.61	101.37

A népszerűségnek sokféle kétes fokmérője van. Ám van egy kétségtelenül elfogadott mérce, s ez a paródia. Ha valamiről paródia készül, akkor azon legalább van mit parodizálni. Így hát nagy örömmel szereztünk róla tudomást, hogy a Műszaki Egyetem Kipufogó című szatírikus kiadványában megjelent egy számítástechnikai melléklet, amelyben bizonyos jelek, megtévesztő stílusjegyek és efféle arra a téves gondolatra engednek következtetni, hogy valami módon mi magunk vagyunk a múzsa. Hogy mi az igazság, győződjenek meg erről saját szemüleg. Olvassák a Kipufogó mellékelt mellékletét!

A KIPUFOGÓ SZÁMÍTÁSTECHNIKAI MELLÉKLETE

HÍROLDAL

Az Országos Oktatástechnikai Központ által kiírt – eredetileg mintegy 50 főre tervezett – tanfolyamra 18 000 egyetemista, közép- és általános iskolai diák jelentkezett. A nagyfokú túljelentkezés oka az, hogy a hirdetés szövegébe sajnálatos hiba csúszott és így jelent meg: „LÓGÓ tanfolyam...''

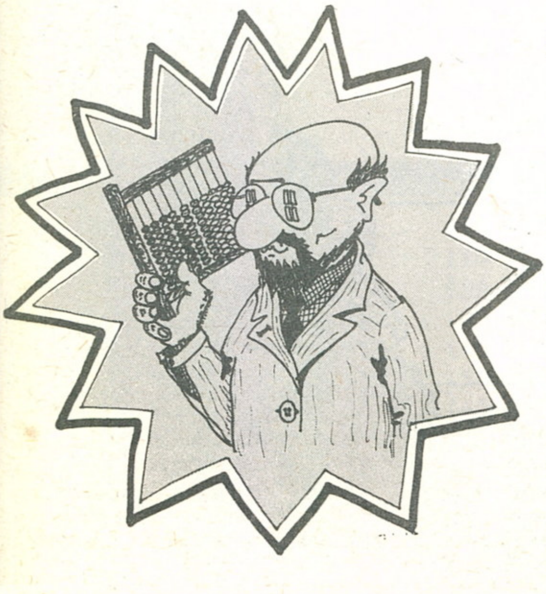
„A keletiek újabb trükkje...'' – kommentálta a Byte magazin az Országos Műszaki Fejlesztésl Bizottság döntését, melyben feloldották a nyugatra irányuló mikroelektronikai exporttilalmat. Az USA kormánya máris a belföldi piac védelmét szolgáló importkorlátozó intézkedéseket fontolgat.

Az I. László Gimnázium Számítógépes Szakkörének tagjai 1984. 03. 25-én 17.38-kor a budapesti telefonhálózaton keresztül egy számunkra ismeretlen számítógépes hálózattal kerültek kapcsolatba. Hazánkban ez volt az első ilyen jellegű illegális behatolás.

„Ó, azok a magyar mikrogépek...!'' – hümmögött elismerően Mr. Clive Sinclair a Sinclair Research Ltd. szellemi atyja, mikor legutóbbi budapesti tanulmányútja alkalmával megismerkedett a HT 00 WC School-computerrel.

A Telefongyár megkezdte a Magyarországon hiánycikként ismert mikronyomtatók sorozatgyártását. Az ütemes termelést nehezíti, hogy a gyár anyagellátási osztálya és a Hungarovin közt még nem született megállapodás a gép egyik legfontosabb alkatrészének, a devizaigényes nyugati importból származó menetes orsónak az elosztását illetően.





VALLATÓ

Tisztelve a hagyományokat, mi is kínpadra vontuk az egyik legelterjedtebb és legnépszerűbb mikroszámítógépet. Mivel a HT–00 School-computer magyar gyártmány, ezt láttuk aktuálisnak kiszálkázni lapunkban. Álláspontunk, hasonlóan testvérlapunkéhoz, a BIT-LET-hez, nekünk is a következő: nem egymáshoz hasonlítjuk a gépeket, hanem azt próbáljuk osztályozni, hogy a saját kategóriájában, a saját áráért mit nyújt egy-egy mikroszámítógép.

KÍNRENDSZER:

A Vallató kiértékelése közben éreztük, hogy a HT–00 sok helyen kilóg a kínrendszer szempontjaiból, nehéz osztályozni! Nem véletlen! Ahogyan a HT konstrukciójú más gépek is kilógnak a világ összes jól bevált, kiforrott szerkezetű gépei közül – úgy feszegeti forradalmi, zseniális újításaival a mi kínrendszerünket is a HT–00. De ez már kiderül a Vallatásból.

Ár

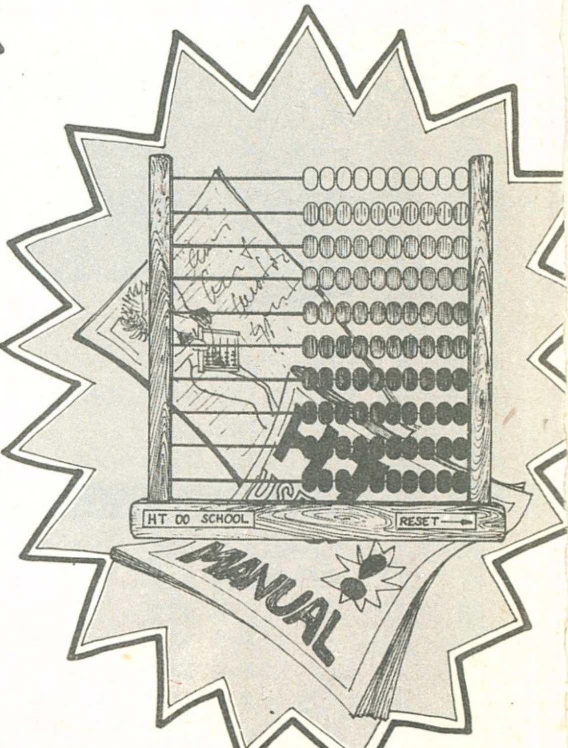
1 KÍN

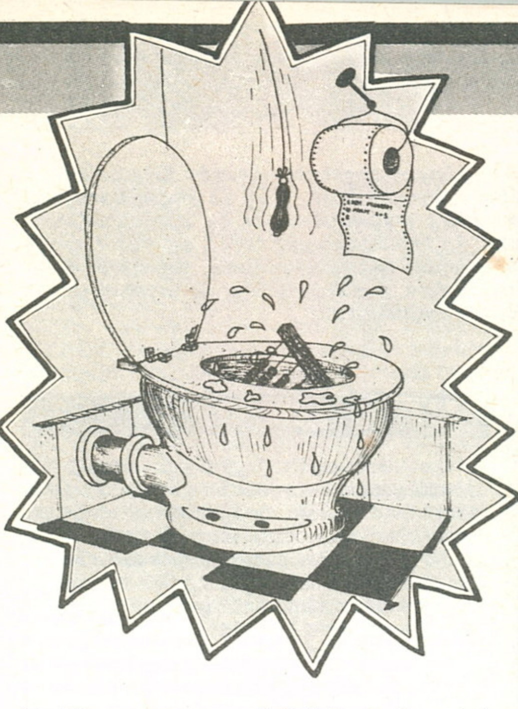
Többféle apró különbséggel bíró fajtáját ismerhettük meg a HT–00 típusnak, de árban (olcsóságban) persze veri a világhírű mikroszámítógépeket. Gyártóink közel ezred részére csökkentették az előállítás költségeit pl. a világhírű ZX 81-hez képest.

Perifériák

2 KÍN

A többféle kivitel elsősorban itt jelentkezik, a HT–00/d pl. cserélhető, színes memóriaegységekkel rendelkező, igen modern kiépítettségű gép. Az alapgéphez több közvetlen csatlakozási lehetőség létezik. Igen sokféle módon illeszthetők a géphez a különböző perifériák. Összeköthető nyomtatóval, lemezegységgel, kazettás magnóval, televízióval. Az összeköttetés megvalósítása is igen sokféle módon történhet, ezt a rendelkezésre álló kötözőanyag határolja le. Történhet damillal (horgászok előnyben), spagival, lánccal stb.





Képernyőkezelés

3 KÍN

A képernyő kezelhetősége rendkívüli, a memóriaegységek megfelelő színkombinációja fantasztikus képek összeállítására alkalmas (a HT–00/d – cserélhető memóriaegységes kivitel, képalkotási variációs lehetőségei rendkívül széles körűek.) A gép kialakításának szintén rengeteg változata létezik: Az új géptulajdonosok kézzel, de a gyakorlottabb kezelők lábbal, orral, sőt nyelvvel is képesek kezelni a gépet. A képernyő háromdimenziós kivitele óriási távlatokat nyit a világtalan számítógépkezelők számára.

Hang

4 KÍN

A gép hangja gyári előállítás után igen egysíkú, a fából készült memóriaegységekből többnyire egyféle kattogó hang nyerhető, de ügyes felhasználó a fa memóriaegységeket könnyűszerrel kicserélheti csont, üveg, fém (ez a legcélszerűbb) részekre, amelyekből már igen szép hangzásokat lehet kihozni. Egy zenei érzékkel megáldott kezelő pl. különböző nagyságú és tömörségű fém memóriaegységekből felépített HT–00-n eljátszotta a Boci-boci tarka kezdetű nótát is.

Kazettás tárolás

5 KÍN

A pótmemóriaegységek tárolására mindenféle (fa, műanyag, fém) kazetta alkalmas. Egy leleményes felhasználó a különböző tároló kazettákat tartalmuknak megfelelő anyagból készítette és megfelelő színűre festette, melyből katalógus nélkül közvetlen úton következtethet a tartalmukra.

Megbízhatóság

6 KÍN

Bátran állíthatjuk, hisz többen úttörő becsület szavukat adták, hogy a HT–00 kezelés közben hamis eredményt még soha nem mutatott gépi hibából eredően. Igaz, egyik olvasónk panaszkodott, hogy egy átprogramozás alkalmával a memóriaegységek százfelé gurultak, de ez minimális kézügyességének tudható be. Azonban – és ez is a HT–00 sokoldalúságát bizonyítja – a fakezűeknek is találtunk megoldást: A memóriaegységek alakja a működést nem befolyásolja, így az ügyetlenebbje használjon kocka alakú memóriaegységeket.

Billentyűzet

7 KÍN

Mint azt már említettük, a géphez csatlakoztatható perifériák rendkívül széles skálán mozognak. Így – bár a gépen billentyűzet nincs – sokféle billentyűzethez kapcsolható. A zongorán pl. a kottatartón remekül elfér, így egy tehetségesebb kétkezű zongorista egyidejűleg működtetheti kedvenc zeneszerszámát és a mikrogépet is.

Dokumentáció

8 KÍN

A gép gyártása Magyarországon igen széles körben folyik, azonban magyar nyelvű szakirodalom a géppel kapcsolatosan még nem jelent meg. Egy-két igen jó nyelvérzékkel megáldott fordítónk értesüléseket szerezhetett az Abakuszocsnaja Gazéta című lapból, mely a Zavód Mehanicsnava Mikrokompjutera nevű gyár szakembereivel tart állandó kapcsolatot. Egyes meg nem erősített források szerint az albán Gojovok Löklek cég többoldalas ismertetőt adott ki a HT–00 típusról, de ezt sajnos ipari kémeinknek nem sikerült beszerezniük. (A kintiek beutazási, a bentiek kiutazási engedélyt nem kaptak.)

Editálás

9 KÍN

Mint tudjuk, ez átszerkesztést, törlést, módosítást jelent. Ezt a HT–00 gyártói igen egyszerű módon oldották meg. A gép, fősíkjában 90 fokkal jobbra vagy balra elfordítva, automatikusan kinullázza az eddig bevitteket. Így a program átszerkesztése azonnal elölről kezdődhet. Egyes szakberkekben a gépet e különleges tulajdonsága következtében a „gravitációs nullázású" jelzővel emlegetik együtt.

Tanulhatóság

10 KÍN

Általában nagyon jól tanulható gépnek tartják, és ezt az is bizonyítja, hogy a szomszéd 3 éves gyermeke, aki ráadásul lány, már most képes egyszerűbb programok futtatására a HT–00-n.

Emberközelség

11 KÍN

Az igen sok dicsérő szó alátámasztja az általános véleményt: a gép igen barátságos jószág, még senkit nem harapott meg. Igaz, egyik felhasználónknak már becsípte az ujját, de az a véletlenek összjátéka eredményeképp következett be, ugyanis hárman nyúltak egyszerre a géphez. És az utóbbi eset is csak emberközelségét bizonyítja, hisz egyszerre hárman is kényelmesen elférnek a közelében.

Szubjektív vélemény

12 KÍN

A főszerkesztő szerint szubjektív véleményünk annyira szubjektív, hogy közlését egyáltalán nem tartja érdemlegesnek.

GÉPNYERŐ

Folytatva hagyományainkat, ezúttal a kínpadra vont HT–00 WC gépek közül kettőnek keressük leendő gazdáját. Mert mi ilyen nagyvonalúak vagyunk.

A feladat a következő:

lapozzon az 5. oldalra, nézze meg az oldalszámot, majd sokáig gondolkodjon rajta. Megfejtésül egy számot kell beküldenie: ha nagyobb, akkor mi nyertünk.

Rajta hát, a felkínált két mikroszámítógép türelmetlenül várja, hogy gazdát cseréljen.

POSTA

„Ham – let" jeligére: Az ön által leírt Yorick fantázianevű RAM-egység kapacitása feltehetőleg GIGABYTE nagyságrendű lehet, de elavult. Ne vegye meg!

A XI. kerületi úttörőknek: Megértjük tanácstalanságotokat. Javaslatunk: az őrs nevét változtassátok meg SOM-ról ROM-ra, az indulótok pedig legyen a CHIP, CHIP, CHÓKA kezdetű dal.

„Biztos rossz!" jeligére válaszunk a következő: Ne keseredjen el, semmi baja a gépnek! Az utasításait akkor sem hajtja végre, ha azokat kiabálva adja, és sajnos látni sem lát, így hiába írja le és tartja elé. A gép nem süket és nem vak. Kezelése a megfelelő programnyelv alapján lehetséges.

„Baj van!" jeligének megnyugtatásul üzenjük: Bizakodjék! Egyszer a férje is meg fogja unni, és akkor ismét önnel tölti az éjszakát. Ez egyébként sem számít házasságtörésnek, főleg ha biztos abban, hogy a számítógépen kívül éjszakánként mással nem foglalkozik a férje.

„Felháborító" jeligére: Felháborodása alaptalan! A BASIC szót angolosan nem úgy kell ejteni.

„Ideges" jeligére: Egyszerűbb, ha kikapcsolja a gépét, amikor valami nem sikerül, mert a GOTO THE BACK SIDE utasítást úgysem tudja értelmezni.



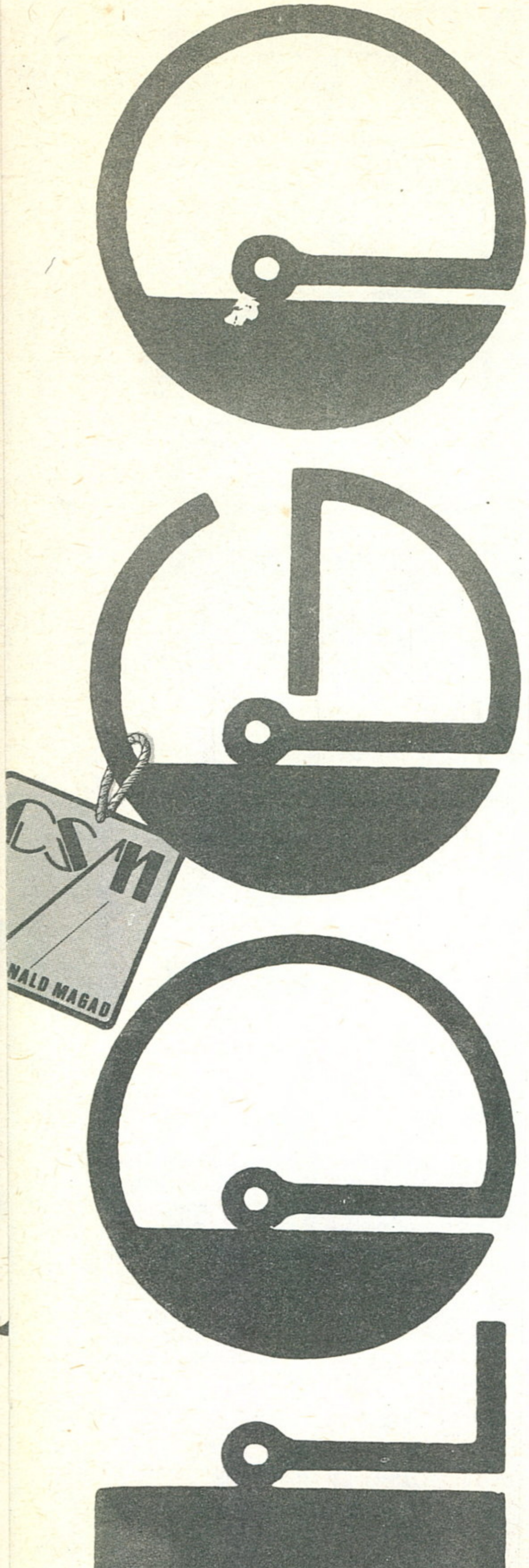
„ZZXX 8811" jeligére: Ön téved! A szellemkép nem a gép hibája. A jelenséget a display mellett álló üres konyakos üveg okozhatja.

„Nem tudok angolul" jeligének azt javasoljuk, hogy legalább a programnyelvet sajátítsa el, ha használni akarja a gépét. Sajnos még a magyar HT 1080Z sem ért magyarul.

„Kíváncsi" jeligének üzenjük: Ne higgyen a barátnőjének, biztos csak ugratni akarja! Olyasmire a legdrágább nyugati gyártmányú mikrogépek sem képesek.

„Félek" jeligére: Ne féljen! Bármennyit is játszik vele, semmiképp nem okoz impotenciát, legalábbis mi nem tudunk a számítógép használatának ilyen következményeiről.

„Manapság a legtöbb iskolában a Számítógéppel segített oktatás kifejezés azt jelenti, hogy a számítógéppel taníttatják a gyerekeket. Úgy is mondhatnánk, hogy a számítógépet használják a gyerek programozására. Az én elképzelésemben a gyerek programozza a számítógépet, és miközben ezt teszi, egyfelől urává válik az egyik legmodernebb és leghatalmasabb technológiának, másfelől bensőséges kapcsolatba kerül a természettudomány, a matematika, az intellektuális modellépítés legmélyebb alapeszméi közül is néhánnyal."
Seymour Papert (MIT)



MIÉRT LOGO?

Amikor végre eldőlt, hogy Magyarországon is számítógépekkel szerelik fel a középfokú iskolákat, rövid és nem túlságosan nyilvános vita folyt arról, hogy a központi program keretében beszerzendő (sőt: gyártandó) iskolaszámítógépek melyik számítógépnyelvet használják. Mint tudjuk, „kiütéssel" győzött a BASIC, melyet valóban a legtöbb személyi számítógépen használnak. Komoly javaslatként merült fel még a PASCAL, azonban ennek hívei végül is vereséget szenvedtek. Szerencsére a számítógépek a beléjük épített nyelven kívül programként bevitt más nyelveken is használhatók. Így ma már egyre több BASIC-alapú mikrogéphez is lehet venni más nyelveket, főleg PASCAL-t, LOGO-t és FORTH-ot. Nem kell tehát lezártnak tekintenünk az iskolákban tanítandó számítógépnyelvek ügyét. A BASIC mellett szólt (elterjedtségén kívül), hogy könnyen megtanulható nyelv. Ez azonban nem pontosan így van. Könnyű a BASIC azoknak, akik már elsajátítottak egyfajta matematikai gondolkodást és elvonatkoztatási készséget. Sok gyerekről ez még a középiskola 3–4. osztályában sem mondható el. Aki megpróbált gyereket tanítani BASIC-re, emlékezhet rá, hogy néhány kiemelkedő képességű gyerektől eltekintve, mennyi nehézségbe ütközik a nyelv elsajátítása, és milyen kevés sikerélményt jelent a tanulása a gyerekek nagy hányadának. Komoly ellenérve a szakembereknek a BASIC-kel szemben, hogy rossz programozási stílusra szoktat (de legalábbis nem szoktat jó stílusra). Mi sem csábítóbb a gyerek (sőt a felnőtt) számára, mint durr bele alapon írni BASIC programot. A rossz struktúrájú programoknál gyakran már egy nap is elég, hogy maga a készítő se értse, mit is akart előző nap. A hátrányosabb helyzetű (szakközépiskola, szakmunkásképző), valamint a fiatalabb (általános iskolás) gyerekek számára ezek a problémák még súlyosabbak. Sajnos, reális a veszély, hogy a gyerekek egy része kudarcként és kényszerként élje át a számítógép-oktatást, és kilépve az iskolából, gyorsan elfelejtse a megemésztetlen, számára idegen tananyagot. Hogy ez milyen szomorú perspektíva (a gyakorlati károkon túl), azt csak az éli át igazán, aki tudja: lehet gyereket – már 4–5 éves kortól! – úgy tanítani számítógép-használatra, hogy az élményt jelent számukra, hogy nem felejtik el, hogy – ha másként nem, gondolkodásmódjukban – egész életükre nyoma marad. De mi lehet a BASIC alternatívája? Sokan ajánlják a PASCAL nyelvet, mely logikus felépítésével, szigorú követelményeivel szinte kényszeríti a programozót az alapos átgondolásra, a jó programozási stílusra. Sajnos, elvontsága miatt a PASCAL-lal ugyanolyan, sőt súlyosabb problémák adódnak a kezdeti szakaszban (és ismét a hátrányos helyzetű, illetve a fiatalabb gyerekeknél is), mint a BASIC-kel. Talán ez is közrejátszott abban, hogy felmerült a kifejezetten pedagógiai célra létrehozandó számítógépnyelv ötlete. Milyen tulajdonságokkal kell egy ilyen nyelvnek rendelkeznie? Segítse elő a problémamegoldó gondolkodás helyes módszereinek kialakulását és rögződését; legyen könnyű az elsajátítása gyerekek számára is, jutalmazza sikerélménnyel a kezdettől fogva; ösztönözze és könnyítse meg a későbbiekben más számítógépnyelvek elsajátítását.

Úgy tűnik, hogy ezeket a célokat eddig leginkább a LOGO számítógép-nyelvet létrehozó munkacsoportnak sikerült elérnie. A LOGO-t a hatvanas-hetvenes években fejlesztették ki Amerikában, a Massachusetts Institute of Technology kutatói, Seymour Papert vezetésével. Ma már számos mikrogépre is kifejlesztették – kapható például az Apple, a Texas Instruments TI 99/4A, a Sinclair Spectrum, a Commodore 64 valamint a TRS 80 változat. Bizonyos alapelvei (és pedagógiai előnyei) pedig egyszerűen „szimulálhatók" bármely nyelven, így BASIC-ben vagy PASCAL-ban is.

A TEKNŐSBÉKA TITKA

A LOGO legszellemesebb újítása a teknősbéka, a teknőc. (Angolul: TURTLE.) A teknőc kedves állat, minden gyerek ismeri, szívesen azonosulnak vele. A gyerek számára a LOGO program kezdetben azt jelenti, hogy egy teknőcnek ad utasításokat, hogy az mit csináljon. Az oktatási laboratóriumokban olyan teknőc is van, mely valóságos guruló szerkezet, és a program szerint ide-oda mozog, vonalat húz maga mögött egy papíron. A kisgépeken azonban a teknősbékát legtöbbször egy nyíl jelzi a képernyőn, és ez mozdul el a program utasításai szerint. Már a négyéves gyerek is megérti és elfogadja, hogy az ott a teknőc orra; és a gimnazista is elfogadja, élvezi a teknőc irányításának lehetőségét. Az orrával jelölt teknőcnek tehát helye van a képernyőn és iránya is van: amerre az orra mutat, arrafelé indul el, ha parancsot kap. Ezen túl, teknőcünknek van egy tolla is, ha elmozdul, „ezzel" egy vonalat húz. A legegyszerűbb LOGO program tehát valahogy így néz ki:

```
ELŐRE 60
```

– és ha ezt beadjuk a gépbe, a teknőc „előremegy 60 teknőclépést". Az eredmény egy 60 egységnyi hosszú egyenes a képernyőn. A teknőc természetesen fordulni is tud. A teknőc-négyzet programja tehát:

```
ELŐRE 60
JOBBRA 90 (ezt fokban adjuk meg)
ELŐRE 60
JOBBRA 90
ELŐRE 60
JOBBRA 90
ELŐRE 60
JOBBRA 90
```

Ez azonban így fantáziátlan és unalmas. Az ISMÉTLÉS utasítással egyszerűsíthetjük:

```
ISM 4
ELŐRE 60
JOBBRA 90
ISM VÉGE
```

Sőt, hogy bármikor bárhova tudjunk négyzetet rajzolni, a teknőcöt megtaníthatjuk a négyzetre (azaz megírjuk a „négyzet" procedúrát, eljárást) és attól kezdve csak elő kell hívnunk a tudományát. A procedúrát az ELJárás szóval kezdjük, nevet adunk neki, hogy máskor is meghívhassuk, és a VÉGE szóval zárjuk le:

```
ELJ NÉGYZET
ISM 4
ELŐRE 60
JOBBRA 90
ISM VÉGE
VÉGE
```

Könnyen belátható a teknőc-mértan alapelve, a „teljes teknőc teoréma": a teknőc akkor tér vissza eredeti helyére és irányába, ha fordulatainak összege 360 fok (vagy annak többszöröse). Egyenlő oldalú sokszögnél tehát a 360 fokot egyszerűen elosztjuk az oldalak számával. Papert egyik gyerekének, Pamelának példája:

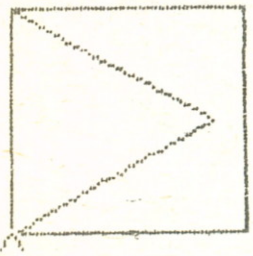
```
ELJ HÁROMSZÖG
ISM 3
ELŐRE 60
JOBBRA 120
ISM VÉGE
VÉGE
```

Pamela azonban házat akart rajzolni, ezért úgy okoskodott, hogy a ház = négyzet+háromszög.

```
ELJ HÁZ
NÉGYZET
HÁROMSZÖG
VÉGE
```

(Azaz: a procedúrák egymáshoz kapcsolásával új procedúrát létesíthetünk. Most már csak a HÁZ-at kell meghívni.)

Pamela programja azonban nem házat rajzol:



Papert hangsúlyozza, hogy a LOGO környezetben a hiba nem baj, nem kudarc, hanem felfedezésszerű élmény: jé, hát EZ lett belőle? Vajon miért? (Aki idáig eljutott az olvasásban, bizonyára tanácsot is tud adni Pamelának, hogy javítsa ki HÁZ eljárását...)

LOGO LOGIKA

Nincs helyünk konkrétan tovább ismertetni a LOGO nyelv részleteit, talán a lényeg ennyiből is világos. BASIC-ben dolgozó olvasóink próbálják meg a házprogramot megírni gépük BASIC-jében! Azonnal rá fognak jönni, hogy a LOGO mennyivel emberközelibb, gyerekközelibb nyelv. Természetesen az itt nem ismertetett parancsaival a LOGO ugyanúgy tudja a matematikai és logikai műveleteket, a feltételes elágazásokat stb. mint a többi számítógépnyelv. Nem játékszer tehát, hanem azokkal egyenértékű magas szintű nyelv, csak éppen mindenkinek megközelíthető nyelvtannal.

A LOGO megközelítésmódja arra tanít, hogy a nagy problémákat bontsuk fel kisebbekre, ezeket oldjuk meg, és illesszük össze a végső megoldássá. (Azaz: létesítsünk procedúrákat és szubprocedúrákat.) Aki LOGO-ban tanul, programozási logikát tanul elsősorban, és csak másodsorban egy konkrét nyelvet. Éppen ezért, a LOGO-n edzett gyerek igen könnyen tanul meg más számítógépnyelvet. A hozzáértő olvasónak feltűnhet a LOGO szerkezeti hasonlósága a PASCAL-hoz: LOGO-ból egyenes út vezet a PASCAL-ba, de a BASIC-be is. Viszont az a tanuló, aki LOGO után lép át BASIC-be, nem úgy fog hozzáfogni a BASIC programhoz, hogy leül és beüti: 10 LET A-0; hanem előbb végiggondolja, hogy mit akar, eljárásokat tervez stb. Nyitva áll előtte az út a strukturált programozáshoz, a maga és mások által is megérthető programokhoz, a jó programozási stílushoz. Reméljük, hogy a fentiekkel sikerült felkelteni pedagógus és gyakorló szülő olvasóink érdeklődését. A továbbiakban megmutatjuk, hogyan lehet a LOGO nyelv néhány érdekességét megvalósítani (szimulálni) a Sinclair Spectrum BASIC-jében.

CSM = CSINÁLD MAGAD

A CSM-LOGO nem versenyezhet a LOGO nyelv profi változataival, de ez nem is célja. Mit tud és mit nem tud? Tudja a teknőcgrafikát, tudja a teknőcöt közvetlen parancsokkal irányítani vagy programmal, ismételhető módon vezérelni. A CSM-LOGO rekurzív, azaz egy-egy eljárás újra „meghívhatja" saját magát. Nagy hátránya a profi változatokhoz képest, hogy lassú, mivel BASIC nyelven írtuk, nem gépi kódban. A BASIC-ben írt programot azonban minden Spectrum-tulajdonos olvasónk megértheti, mert nemcsak a listát közöljük, hanem azt is, hogyan működik. CSM-LOGO változatunkban nincsen listafeldolgozás, ami az igazi LOGO-ban alapvető. De egyelőre, az első változatban nincsen benne feltételes elágazás sem (lásd a BASIC-ben: IF-THEN), nincsenek változók és nem végez számtani-logikai műveleteket sem. Miért írtuk, hogy egyelőre? Azért, mert CSM-LOGO-nk építőkocka elven épül (bikkfanyelven: moduláris szerkezetű), azaz minden olvasónk saját ízlése és szükségletei szerint bővítheti. Az első változat tehát csak az alap, belefér a 16 K-s Spectrum tárolójába is. A bővített változatokhoz (ld. később) már a 48 K-s memóriára lesz szükség.

A CSM–LOGO SZERKEZETE

Ha végiggondoljuk, mit kell „tudnia" nyelvprogramunknak, többé kevésbé világosan előttünk áll kívánatos szerkezete is. Kell egy bemeneti rész, ahol PARANCSAINKAT (azonnali végrehajtás) vagy ELJÁRÁSAINKAT (későbbi végrehajtás külön parancsra) beadhatjuk a gépbe. Az eljárásokat a gépnek tárolnia is kell, hiszen többször akarjuk használni őket. A CSM-LOGO az eljárásokat a P$ nevű string változóban tárolja. De tárolnunk kell az eljárások neveinek „tartalomjegyzékét" is, tehát hogy melyik eljárás hol található a P$ stringen belül. A bemeneti részt bohókás nagyravágyással a komoly gépekhez hasonlóan mi is SZERKESZTŐ-nek (magyarul editor...) fogjuk nevezni.

Parancs-programjainkat és eljárás-programjainkat a gépnek le kell fordítania a maga számára. Ha azt parancsoljuk: ELŐRE 60, a gépnek „értelmeznie" kell, hogy az ELŐRE parancs nyomán rajzolnia kell egy egyenes vonalat, és a „60" azt jelenti, hogy ez a vonal 60 egység hosszú lesz. Készül tehát egy FORDÍTÓ programrészünk (interpreter). Ez három alrészből épül fel: a BEHATÁROLÓ megkeresi a programban a szavak – számok határait, majd ha talált egy szót, akkor továbbadja azt azonosítónak. Az AZONOSÍTÓ megkeresi, hogy hányadik utasításról vagy eljárásról van szó. (Ha nem találja, közli a használóval, hogy nem ismeri a... parancsot). Végül a költői nevű VÉGREHAJTÓ elvégzi, amit a parancs kíván, és azután visszalép a BEHATÁROLÓ részhez a következő végrehajtandó parancsért. Ha ilyen nincs (azaz VÉGE volt) akkor visszatérünk a SZERKESZTŐ részbe, várva az új tennivalókat.

A CSM-LOGO grafikai célú programnyelv. Kell tehát bele egy olyan rész, mely könnyen és egyszerűen rajzol. Fentebb, a 3. részben azzal cukkoltuk a BASIC-ben dolgozó kollégákat, hogy próbálják csak meg a ház-rajzoló programot megírni BASIC-ben. Most leleplezzük önmagunkat: a CSM-LOGÓ-ban is Sinclair BASIC utasítások fogják megrajzolni a megrajzolnivalókat, „csak" éppen ezeket előre beépítjük a CSM-LOGO nyelvbe, tehát a használónak a részletekkel már nem kell törődnie. Lesz tehát a programnak egy RAJZOLÓ része is. Ennek lényege a VONALHÚZÓ, és csatlakozik hozzá a NYÍLRAJZOLÓ és a NYÍLTÖRLŐ. Teknőcünknek ugyanis orra van (egy kis nyílhegy), ebből látjuk, HOL van és MERRE néz. Ha a teknőcöt elindítjuk valamerre, akkor ezt az orrot le kell törölnünk a képernyőről, és a megérkezés után az új helyen és új irányban ismét ki kell rajzolnunk. Végül általában bármiféle programnak kell egy INDÍTÓ rész, ahol a változók megkapják kezdő értékeiket, a tömbváltozókat kijelöljük (deklaráljuk), stb. A CSM-Logóban kétféle indításunk lesz: a HIDEGINDÍTÓ üresen indul, a MELEGINDÍTÓ viszont megtartja addig írott LOGO-eljárásainkat. A MELEGINDÍTÓ a HIDEGINDÍTÓN belül egy rész

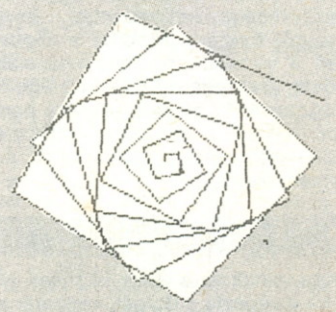
```
ELJ SZIROM
  ELORE 30
  JOBBRA 163
  SZIROM
VEGE
```



```
ELORE 40
SZIROM
```

```
ELJ SPIRAL  :OLDAL  :SZOG  :UJRA
  ORRNE
  ISM  :UJRA
    ELORE  :OLDAL
    JOBBRA  :SZOG
  LEGYEN  :OLDAL  :OLDAL + 3
  ISMVEGE
VEGE

VALTOZOK SPIRAL
```



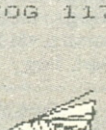
```
ELJ SPIRAL  :OLDAL  :SZOG  :UJRA
  ORRNE
  ISM  :UJRA
  ELORE  :OLDAL
  JOBBRA  :SZOG
  LEGYEN  :OLDAL  :OLDAL + 2
  ISMVEGE
VEGE

ELJ VALTOZOK  :OLDAL  :SZOG  :UJRA
  LEGYEN  :OLDAL 3
  LEGYEN  :SZOG 90
  LEGYEN  :UJRA 20
VEGE

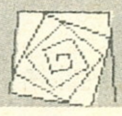
VALTOZOK SPIRAL
```



```
VALTOZOK
LEGYEN  :SZOG 117 SPIRAL
```



```
VALTOZOK
LEGYEN  :SZOG 85 SPIRAL
```



A CSM–LOGO FELÉPÍTÉSE:

RAJZOLÓ
NYÍLTÖRLŐ
VONALHÚZÓ
NYÍLRAJZOLÓ
FORDÍTÓ
BEHATÁROLÓ
AZONOSÍTÓ
VÉGREHAJTÓ
SZERKESZTŐ
(A VÉGREHAJTÓ EGYES SZUBRUTINJAI)
INDÍTÓ
HIDEGINDÍTÓ
MELEGINDÍTÓ
(Ha a CSM-LOGO-t már LOGO-ban írhattuk volna, ezek lennének az eljárások...)

A CSM–LOGO I. VÁLTOZATA

E sorok írója meglehetősen utálja a programsorok mechanikus „bepötyögését" a számítógépbe. Talán más is így van ezzel. Ezért a CSM-LOGO programot ugyan közöljük elejétől végéig, de javasoljuk, hogy sorait NE egytől tízezerig írják be a gépbe, hanem abban a sorrendben, ahogy az alábbiakban a szöveg javasolja. Igyekszünk úgy felépíteni a programot, hogy már nagyon hamar „csináljon valamit". A kezdeti primitív szerkezetet azután fokozatosan gazdagítjuk a LOGO-tudománnyal.

A SZERKESZTŐ RÉSZ

A programnak ez a része végtelen hurkot alkot: ha elvégezte a dolgát, visszatér a kezdethez, új feladatot vár, nem áll le. (1000, 1190).
Az A$ stringbe gyűjtjük először a beadott programsorokat, és a SORSZ változó tartalmazza azt a SORSZámot, ahová a képernyőn a következő programsort kiíratjuk. (1010, 1020).
A program sorait a B$ string tárolja (1100), amíg be nem kerülnek a program P$-jába. Ha ki akarunk lépni a LOGO végtelen hurkából BASIC-be, akkor S-et adunk be. (1105). Ha üres stringet adtunk be, azaz csak megnyomtuk az ENTER billentyűt, soremelés történik. (1110).
Az 1120 szóközt tesz utána, az 1142 az éppen beadott sort a többihez adja. Az 1135 helyet csinál a képernyőn, az 1140 kiírja a LOGO programsort és „sort emel".
Ha VÉGE parancsról van szó, akkor továbbengedi a vezérlést, de ha nincs még vége, és nem is eljárásról van szó, akkor az 1144 visszaküldi 1100-ba. Az 1130, 1133, 1137, 1138 sorok a kiírt programképet szebbé teszik.
Az 1141. sor levágja a B$ elejéről a (fölösleges) szóközöket. Ha ELJárást írtunk, az 1125 sor beállít egy SORHOSSZ változót, parancsnál erre nincs szükség, mert azt nem tároljuk. Ha egy eljárás utolsó sorát beírtuk, az 1150 sor boldogan felismeri, hogy VEge az eljárásnak, és intézkedik.
Folytassuk a PARANCS-kezelő résszel. (1300)
A 1310 elteszi a P$ végének címét későbbre. Azonban a P$-t még nem deklaráltuk! El kell tehát kezdenünk az INDÍTÓ rész (9000-től) kidolgozását. (9010, 9080)
A következők megértéséhez néhány szót szólnunk kell a „verem" működési elvéről.

A VEREM

A CSM-LOGO verme hasonlóan működik, mint a Sinclair BASIC-beli GOSUB utasításé. Érdemes tehát elolvasni a gépkönyv 5. fejezetét! Röviden arról van szó, hogy a verem „legtetején", azaz az éppen legnagyobb sorszámú változóban a V ( ) tömbváltozón belül (9030) ott őrzi a gép a KÖVETKEZŐ elvégzendő feladatának CÍMÉT, azaz a helyét a P$ programtárolóban. A VM változó (a veremmutató) mondja meg, hogy éppen melyik sorszámú az érvényes cím. Ha a gép ELVÉGZETT egy feladatot, akkor mintegy kipipálja, azaz leveszi a címet a verem tetejéről. Ilyenkor a VM értékét eggyel csökkenti. Ha viszont új feladatba fog, mielőtt a régit befejezte volna, ráteszi a verem tetejére az új címet, és eggyel növeli a VM értékét. Világos? (Nem szégyen bevallani, ha még nem világos. Reméljük, hogy a FORDÍTÓ rész működésének leírása során ez is érthetővé válik.)
Az PARANCS részhez tehát írjuk hozzá az 1320, 1330, 1340, 1350 és 1390 sorokat. Az 1305. sor célja később derül ki. Építőkocka elvünknek megfelelően most arra törekszünk, hogy CSM-LOGO programunk mielőbb működjék – valahogy. Ezért az ELJÁRÁS rész kidolgozását későbbre hagyjuk, és csak az 1700 sort írjuk be, valamint egy biztonsági sort: 1710 RETURN
A fordítás kezdete: a behatároló
100 REM FORDITO
110 IF VM = 0 THEM RETURN: REM Ha nincs feladatcím a veremben, akkor nincs mit csinálni, vissza kell térni a Szerkesztő-be.
200 REM BEHATÁROLÓ – Itt keressük meg az utasítások végét a programban.
205 LET I$ = " "
210 LET V$ = "NEMVEGE": REM A Vége-utasítás alapvetően különbözik a többitől: Utána már nincs mit tenni. Ezért a V$ lesz a „VÉGE-zászló", itt „felcsapjuk". Mint látni fogjuk, a 230. sor mindig megvizsgálja, „le van-e csapva".
220 FOR I = V(VM) TO LEN P$
240 NEXT I
Ez a FOR-NEXT hurok a verem tetején megadott kezdőcímtől kezdve a program végéig (amit a LEN P$ mutat) végig megvizsgálja a P$ tartalmát. Szóközt keres, mert az a tagolóelemünk. (Ld. a 230 sort.)
Ezért olyan fontos a CSM-LOGO nyelvtanában, hogy minden utasítás vagy számjegy (paraméter) után tegyük ki a szóközt.
A FOR–NEXT hurokban a 230 sor tehát addig „keres", amíg szóközre nem lel. Ha talált szóközt, akkor már tudja: ami az indulás és a szóköz között van, az a keresett szó. Ezt az ippeg meglelt szót beteszi az I$ stringváltozóba.
A 230 sor megértéséhez tegyük fel, hogy a következő CSM-LOGO programot adtuk be a P$-ba:
ELORE 60 VEGE
1234567891234
Amint látjuk, ez a P$ a végén lévő kötelező szóközzel együtt 14 betűnyi hosszú. Ez esetben a veremmutató VM = 1 és az általa jelölt első veremhelyen V(1)=1, mivel a feladat a legelső betűnél kezdődik, az E-nél. A 220-as FOR-NEXT tehát I=1-nél kezd, és I=6-nál találja meg az első szóközt. Most működésbe lép a 230-as sor, és az I$ értéke a P$ 1 és I–1 közötti, azaz 1 és 5 közötti része lesz. Azaz I$ egyenlő lesz az "ELORE" stringgel. A verem tetején az értéket most lecseréljük I+1-re, azaz 7-re, mivel a következő feladat a 7-nél kezdődő„ valami" értelmezése lesz. A vezérlés most átkerül az AZONOSÍTO részbe, és ha onnan visszajövet a V$ tartalma az lesz, hogy "VEGE", akkor visszamegy a FORDÍTO elejére. Ott kiderül, hogy van-e még feladatcím a veremben. Ha nincs, azaz ha VM=0, akkor vissza lehet érni a Szerkesztőbe új feladatért.
290 STOP: REM. Ha minden rendben lesz, erre a sorra sose lesz szükség. De ha mégsem, akkor hibás program nem fut bele valami másba, hanem itt megáll. Hogy a rák a vetésre ne menjen...
Az utasítás-AZONOSÍTÓ
500 REM AZONOSITO (520, 550, 560, 565.)
550 LET I$=I$+"–7 szóköz–":LET I$=I$(TO 7):REM I$ hosszát 7-re szabjuk, hogy összehasonlítható legyen a 7 betűs hosszúságú fenntartott szavainkkal
560 FOR A=1 TO 30: IF I$=F$(A) THEN GO SUB (700+A):RETURN:REM Ha teljesül, akkor fenntartott szót talált, és elindul a VÉGREHAJTÓ részbe.
565 NEXT A:REM Ha a 13. értéknél sem talál, akkor tovább keres az ELJÁRÁSOK között
570 REM ELJÁRÁSOK – egyelőre kimaradnak
595 PRINT AT 20,0: "Nem ismerem a ";I$; parancsot!":LETV$="VEGE":LET VM=0:RETURN:REM Ha nem találta meg sehol, akkor ezt közli.
Az utasítás-VÉGREHAJTÓ
E rész feladata, hogy a fentiekben beazonosított elemi CSM-LOGO utasításokat BASIC-ben végrehajtsa. Az 560 sor küldi a megfelelő helyre a vezérlést.
700 REM VÉGREHAJTÓ (és a 701–729 sorok.)
A FOK-változó a teknős irányát tárolja. A 10–70. sorok BASIC szubrutinjait (nyíltörlő, vonalhúzó, nyílrajzoló), mivel elvi jelentőségük csekély, helyhiány miatt nem kommentáljuk.
Felhívjuk a figyelmet a 705–706–707 sorokra, amelyek az ISMétlést végzik. Itt ugyanolyan veremszerkezettel találkozunk, mint korábban az utasítások végrehajtásánál. A veremmutatót és számlálót a 9550, 9560 és 9570 sorok deklarálják. Ahhoz, hogy használatba-vehessük programunkat, már csak a 9000–9800 sorokat kell beütnünk, illetve amelyik még hiányzik közülük. (Ez az INDÍTÓ blokk.) Mielőtt elindítanánk RUN 9000-rel, okvetlenül mentsük ki GO TO 9800-zal és ellenőrizzük a kimentést. CSM-LOGOnk utasításait a 9060. sorban láthatjuk, de ne feledjük: egyelőre még nem mind működik! De a korábban leírt négyzetrajzolást már kipróbálhatjuk. Az egyes sorokat külön-külön adjuk be, minden sor után megnyomjuk az ENTER billentyűt.
A CSM-LOGO most még mindent azonnal végrehajt, tárolható programot nem használ. Nem ismeri fel az ELJárásokat. A következőkben lépésenként fejlesztjük tovább a programot, hogy tudjon többet és legyen kényelmesebb.

A II. VÁLTOZAT: ELJÁRÁSOK

Most hozzákezdhetünk a tulajdonképpeni ELJÁRÁS szubrutin kidolgozásához. Az első sorunk, a 1700 REM ELJARAS már megvan. A rákövetkező, 1710-es sort természetesen töröljük, illetve újat (1720) írunk helyette, amely megkeresi az eljárásra utaló első szó (az "ELJARAS" vagy rövidítve "ELJ") utáni első szünetet. Ha megvan, ezt a fölösleges részt „levágjuk" az A$ munkaváltozónk elejéről és továbbmegyünk 1730-ra.
Ez bizony bonyolult sor, de igen fontos! Először megkeresi ismét a szóközt, és ezáltal behatárolja az eljárás nevét. Ezt a nevet beteszi a B$ munkaváltozóba. Az eljárás címe az E tömbváltozó soronkövetkező (ld. ELJSZAMA) rekeszébe kerül. Figyeljük meg, hogy a cím az eljárás neve utáni első LOGO-parancs első betűjének címe! Az eljárás neve ugyanis nem parancs, a LOGO parancsként

LOGO

```
   1 REM CSM-Logo/7.
   2 REM
   3 REM HA-eljarassal. A "LEgye
n" helyett  "ELJaras" irando!
   4 REM
   5 REM © SZEKFU ANDRAS 1984
   6 REM
   7 REM SOLARSOFT - INTEGRAL
   8 REM
   9 GO TO 9000
  10 REM NYILTORLO
  11 IF ORR=0 THEN RETURN
  12 LET XP=COS (OLDRAD-2.5): LE
T YP=SIN (OLDRAD-2.5)
  13 PLOT X,Y
  14 DRAW INVERSE 1;XP*5,YP*5
  16 PLOT X,Y
  17 LET XP=COS (OLDRAD+2.5): LE
T YP=SIN (OLDRAD+2.5)
  18 DRAW INVERSE 1;XP*5,YP*5
  19 RETURN
  20 REM VONALHUZO
  21 LET RAD=DG*PI/180
  22 LET XP=COS RAD: LET YP=SIN
RAD: REM XPLUSZ,YPLUSZ
  23 IF PP=1 THEN PLOT X,Y
  24 IF PP=0 THEN PLOT INVERSE 1
;X,Y
  28 IF X+XP*TL>250 OR X+XP*TL<5
 THEN  LET TL=TL-SGN TL: BORDER
0: BORDER 7: GO TO 28
  29 IF Y+YP*TL>170 OR Y+YP*TL<5
 THEN LET TL=TL-SGN TL : BORDER
0: BORDER 7: GO TO 29
  30 IF PP=1 THEN DRAW XP*(TL-1)
,YP*(TL-1)
  40 LET X=X+XP*TL: LET Y=Y+YP*T
L
  42 RETURN
  45 IF ORR=0 THEN RETURN : REM
NYILRAJZOLO
  46 LET RAD=DG*PI/180
  47 PLOT X,Y
  50 LET OLDRAD=RAD
  52 LET XP=COS (OLDRAD-2,5): LE
T YP=SIN (OLDRAD-2.5)
  54 DRAW XP*5,YP*5
  56 PLOT X,Y
  58 LET XP=COS (OLDRAD+2.5): LE
T YP=SIN (OLDRAD+2.5)
  60 DRAW XP*5,YP*5
  70 RETURN
 100 REM FORDITO
 110 IF VM=0 THEN RETURN
 200 REM BEHATAROLO
 205 LET I$=""
 210 LET V$="NEMVEGE"
 220 FOR I=V(VM) TO LEN P$
 230 IF P$(I)=" " THEN LET I$=P$
(V(VM) TO I-1): LET V(VM)=I+1: G
O SUB AZONOSITO: IF V$="VEGE" TH
EN GO TO FORDITO
 240 NEXT I
 290 STOP
 300 REM VALTOZOK ES PARAMETEREK
 310 DIM P(6): REM PARAMETEREKNE
K URESEN
 320 LET PAR=1: REM HANYADIK PAR
.KOVETKEZIK?
 330 IF LEN P$<I+1 THEN RETURN
 340 IF (P$(I+1)<=":" AND P$(I+1
)>"/") THEN GO TO 344: REM VALTO
ZO VAGY SZAM
 342 IF PAR>1 THEN LET PAR=PAR-1
: RETURN : REM NINCS PAR.
 343 RETURN
 350 GO SUB VALTOZO: IF P$(I+1)=
":" THEN GO TO 350 : REM 800
 360 IF PAR>3 THEN RETURN : REM
TOBB PAR. MAR NEM LEHET
 370 REM SZAMKERESO
 374 LET C$=""
 376 IF LEN P$<=I THEN RETURN
 380 IF (P$(I+1)<":" AND P$(I+1)
>"/") THEN LET C$=C$+P$(I+1): LE
T I=I+1: GO TO 380: REM NUMBER(P
ARAMETER)
 385 IF C$<>"" THEN LET P(PAR)=V
AL C$: LET V(VM)=I+2: LET I=I+1:
 LET PAR=PAR+1: REM PARAMETER ER
TEKE
 390 GO TO 330
 500 REM AZONOSITO
 520 IF I$="(" OR I$=")" THEN RE
TURN
 550 LET I$=I$+"        ": LET I
$=I$( TO 7)
 560 FOR A=1 TO 30: IF I$=F$(A)
THEN GO SUB 700+A: RETURN : REM
FENNTARTOTT SZO
 565 NEXT A
 570 REM ELJARASOK
 580 FOR A=1 TO ELJSZAMA: IF I$=
E$(A, TO 7) THEN LET VM=VM+1: LE
T V(VM)=E(A,1): LET V$="VEGE": R
ETURN : REM ELJARAST TALALT
 585 NEXT A
 595 PRINT AT 20,0;"NEM ISMEREM
A ";I$;" PARANCSOT!": LET V$="VE
GE": LET VM=0: RETURN
 700 REM VEGREHAJTO
 701 GO SUB PARVALT: LET TL=P(1)
: GO SUB 10: GO SUB 20: GO SUB 4
5: RETURN : REM ELORE
 702 GO SUB PARVALT: LET TL=-P(1
): GO SUB 10: GO SUB 20: GO SUB
45: RETURN : REM HATRA
 703 GO SUB PARVALT: GO SUB 10:
LET DG=DG-P(1): GO SUB 45: RETUR
N : REM JOBBRA
 704 GO SUB PARVALT: GO SUB 10:
LET DG=DG+P(1): GO SUB 45: RETUR
N : REM BALRA
 705 GO SUB PARVALT: LET RSP=RSP
+1: LET R(RSP)=I+1: LET W(RSP)=P
(1)-1: RETURN : REM ISMETLES
 706 IF W(RSP)>0 THEN LET I=R(RS
P): LET V(VM)=I: LET W(RSP)=W(RS
P)-1: RETURN : REM ISMVEGE
 707 LET RSP=RSP-1: RETURN : REM
 ISMETLES BEFEJEZVE
 708 LET V$="VEGE": LET V(VM)=0:
 LET VM=VM-1: RETURN : REM VEGE
 709 GO SUB 2200: RETURN : REM
LEGYEN (ERTEKADAS)
 710 GO SUB PARVALT: PRINT P(1):
 RETURN : REM IRD
 711 LET PAR=2: GO SUB 330: LET
P(1)=P(1)+P(2): LET P(2)=0: LET
PAR=1: GO SUB 2230: RETURN : REM
 +
 712 LET PAR=2: GO SUB 330: LET
P(1)=P(1)-R(2): LET P(2)=0: LET
PAR=1: GO SUB 2230: RETURN : REM
 -
 713 LET PAR=2: GO SUB 330: LET
P(1)=P(1)*P(2): LET P(2)=0: LET
PAR=1: GO SUB 2230: RETURN : REM
 *
 714 LET PAR=2: GO SUB 330: LET
P(1)=P(1)/P(2): LET P(2)=0: LET
PAR=1: GO SUB 2230: RETURN : REM
 /
 715 LET ORR=1: GO SUB 45: RETUR
N : REM ORRAT KIDUGJA
 716 GO SUB 10: LET ORR=0: RETUR
 : REM ORRAT BEHUZZA
 717>GO SUB 2000: RETURN : REM P
ROGRAM
 718 GO SUB 2100: RETURN : REM C
IMEK (ELJARASOKE)
 719 GO SUB PARVALT: INK P(1): G
O SUB 45: RETURN : REM TINTA
 720 GO SUB PARVALT: PAPER P(1):
 CLS : GO SUB 45: RETURN : REM P
APIR
 721 GO SUB PARVALT: GO SUB 2300
: LET PAR=2: GO SUB 330: GO SUB
2400: RETURN : REM HA
 722 GO SUB 9510: RETURN : REM K
OZEP
 723 CLS : FOR A=1 TO 30: PRINT
F$(A): NEXT A: PRINT "
           TOVABB> "; FLASH 1;"*"
: PAUSE 0: CLS : GO SUB 45: RETU
RN : REM SOS
 724 LET PP=1: RETURN : REM TOLL
 725 LET PP=0: RETURN : REM TOLL
NE
 726 CLS : GO SUB 45: RETURN : R
EM SZIVACS
 727 GO SUB 9000: RETURN : REM F
ELEJT
 728 GO SUB 2200: RETURN : REM
LEGYEN (ERTEKADAS)
 729 GO SUB PARVALT: BORDER P(1)
: RETURN : REM KERET
 800 REM VALTOZOK KEZELESE
 810 IF P$(I+1)<>":" THEN RETURN

 812 LET VKEZD=I+2: LET I=I+2
 814 IF P$(I)<>" " THEN LET I=I+
1: GO TO 814
 816 LET B$=P$(VKEZD TO I-1): LE
T B$=B$+"        ": LET B$=B$( TO
 7)
 820 LET B$=B$+"        ": LET B$
=B$( TO 7)
 825 LET V(VM)=I+1: REM A VERMET
 IS AT KELL TENNI A VALT.MOGE!
 828 IF PAR>3 THEN RETURN : REM
TOBB VALTOZO MAR NEM LEHET
 830 FOR B=1 TO ELJSZAMA
 840 FOR C=1 TO VAL E$(B,8)
 841 LET VNCIM=C*7+2
 842 IF B$=E$(B,VNCIM TO VNCIM+6
) THEN LET P(PAR)=E(B,C+1): LET
PAR=PAR+1: GO TO 880: REM MEGTAL
ALTA ES KERESI A KOV.VALTOZOT
 850 NEXT C
 860 NEXT B
 890 RETURN
```

Épits ki olyan rendszert,

amelyet hülyék is használni tudnak –

csak hülyék fogják használni!

(A Shaw-elv)

```
 999 STOP
1000 REM SZERKESZTO
1010 LET SORSZ=0
1020 LET A$=""
1100 INPUT "LOGO:";B$
1105 IF B$="S" THEN STOP
1110 IF B$="" THEN LET SORSZ=SOR
SZ+1: GO TO 1100
1120 IF B$(LEN B$)<>" " THEN LET
 B$=B$+" "
1125 IF B$( TO 3)="ELJ" THEN LET
 SORHOSSZ=LEN B$
1130 LET TAB=0
1133 IF SORSZ>20 THEN LET SORSZ=
0
1135 PRINT AT SORSZ,0;"
"
1137 PRINT AT SORSZ+1,0;"
"
1138 IF B$( TO 3)<>"ELJ" AND B$(
 TO 2)<>"VE" THEN LET TAB=TAB+2
1140 PRINT AT SORSZ,TAB;B$: LET
SORSZ=SORSZ+1: FOR A=1 TO (LEN B
$-2): IF B$(A)=" " THEN LET B$=B
$(2 TO ): NEXT A
1141 IF B$(1)=" " THEN LET B$=B$
(2 TO ): GO TO 1141
1142 LET A$=A$+B$
1144 IF A$( TO 3)<>"ELJ" THEN LE
T A$=A$+"VEGE ": GO SUB PARANCS:
 LET A$="": GO TO 1100
1160 IF B$(1 TO 2)="VE" THEN GO
SUB ELJARAS: GO TO 1000: REM ELJ
ARAS=1700
1160>GO TO 1100
1190 GO TO SZERKESZTO: REM 1000
1300 REM PARANCS
1305 LET RSP=0: REM ISM.VEREMMUT
1310 LET PRVEGE=LEN P$: REM ELTE
SZI A PROGRAM-VEGE CIMET
1320 LET VM=1: LET V(VM)=PRVEGE+
1: REM A VEREMBE ELTESZI A PARAN
CS CIMET,AZ 1.HELYRE
1330 LET P$=P$+A$
1340 GO SUB 100: REM FORDITO
1350 LET P$=P$( TO PRVEGE): REM
LEVAGJA A PARANCSOT,MERT AZT NEM
 TAROLJUK
1390 RETURN : REM VISSZA A SZERK
ESZTOBE
1700 REM ELJARAS
1720 FOR A=4 TO LEN A$: IF A$(A)
=" " THEN LET A$=A$(A+1 TO ): LE
T SORHOSSZ=SORHOSSZ-A: GO TO 173
0
1722 NEXT A: REM LEVAGJA AZ "ELJ
ARAS"-T ES LEVONJA A HOSSZBOL
1724 RETURN : REM EZ CSAK NYELV-
TANI HIBA ESETERE KELL!
1730 FOR A=1 TO LEN A$: IF A$(A)
=" " THEN LET B$=A$( TO A-1): LE
T E(ELJSZAMA,1)=LEN P$+SORHOSSZ+
1: LET B$=B$+"       ": LET B$=B
$( TO 7): LET E$(ELJSZAMA, TO 7)
=B$: GO TO 1740: REM MEGTALALTA
AZ ELJARAS NEVET ES ELTETTE
1732 NEXT A
1734 RETURN : REM EZ CSAK NYELV-
TANI HIBA ESETERE KELL!
1740 LET P$=P$+A$: LET PRVEGE=LE
N P$
1750 IF A$(A+1)<>":" THEN LET EL
JSZAMA=ELJSZAMA+1. RETURN
1760 LET VKEZD=A+1
1762 LET VSZAMA=VAL E$(ELJSZAMA,
8)
1764 LET VNCIM=9
1770 FOR A=VKEZD TO LEN A$
1780 IF A$(A)=":" THEN GO SUB 18
00: GO SUB 1900: REM VALTOZOT TA
LALT
1785 NEXT A
1790 LET ELJSZAMA=ELJSZAMA+1: RE
TURN
1800 REM HA VALTOZOT TALALT:
1810 LET VKEZD=A+1
1820 FOR B=VKEZD TO LEN A$-1: IF
 A$(B)=" " THEN RETURN : REM MEG
TALALTA A NEV VEGET
1830 NEXT B
1840 RETURN
1900 REM ELHELYEZI A MEGLELT VNE
VET
1910 LET VVEGE=B-1: LET B$=A$(VK
EZD TO VVEGE): LET B$=B$+"
": LET B$=B$( TO 7)
1920 IF VNCIM+6>29 THEN RETURN :
 REM TELE AZ E$
1930 LET E$(ELJSZAMA,VNCIM TO VN
CIM+6)=B$: LET VSZAMA=VSZAMA+1:
LET E$(ELJSZAMA,8)=STR$ (VSZAMA)
: LET VNCIM=VNCIM+7: LET A=VVEGE
1940 RETURN
1990 STOP
2000 REM PROGRAM
2010 CLS
2015 LET B=1
2020 FOR A=1 TO LEN P$
2030 PRINT P$(A);
2040 IF B=30 THEN PRINT : PRINT
"123456789012345678901234567890"
: PRINT : LET B=0
2050 LET B=B+1
2060 NEXT A
2070 PRINT : PRINT "123456789012
34567890123456789 0": RETURN
2100 REM ELJARASOK LISTAJA
2110 CLS : PRINT "ELJ.CIME","NEV
E": PRINT
2120 FOR A=1 TO 20
2130 PRINT E(A,1);" ";E$(A)
2135 PRINT E(A,2);" ";E(A,3);" "
;E(A,4)
2140 NEXT A
2150 RETURN
2200 REM ERTEKADAS (LEGYEN=MAKE)
2202 REM IF PAR>3 THEN RETURN :
REM NINCS HELY TOBB PAR.NAK
2205 LET I=I+1: LET VKEZD=I+1
2210 IF P$(I)<>" " THEN LET I=I+
1: GO TO 2210: REM MEGKERESI A
PAR. ELEJET
2215 LET VVEGE=I-1: LET D$=P$(VK
EZD TO VVEGE): LET D$=D$+"
": LET D$=D$( TO 7)
2220 GO SUB PARVALT: REM PARAMET
ERT KERES
2230 FOR B=1 TO ELJSZAMA
2235 FOR C=1 TO VAL E$(B,8)
2240 LET VNCIM=C*7+2
2250 IF D$=E$(B,VNCIM TO VNCIM+6
) THEN LET E(B,C+1)=P(PAR): RETU
RN
2260 NEXT C
2270 NEXT B
2280 PRINT AT 21,0;D$;" VALTOZO
NINCS DEKL.!"
2290 RETURN
2300 REM HA-RELACIO KERESO
2310 FOR J=V(VM) TO LEN P$
2320 IF P$(J)=" " THEN LET H$=P$
(V(VM) TO J-1): LET V(VM)=J+1: L
ET I=J: RETURN
2330 NEXT J
2340 PRINT "HA UTAN HIANYZIK A R
ELACIO!"
2350 RETURN
2400 REM HA-ZAROJELKERESO
2410 FOR K=V(VM) TO LEN P$
2420 IF P$(K)=")" THEN LET ZCIM=
K: GO TO 2450
2430 NEXT K
2440 PRINT "A ) HIANYZIK!": RETU
RN
2450 REM HA-MUVELET
2460 IF VAL (STR$ P(1)+H$+STR$ P
(2))=0 THEN LET V(VM)=ZCIM+2: LE
T I=ZCIM+2
2470 RETURN
7990 STOP
9000 REM INDITAS
9005 POKE 23658,24: REM NAGYBETU
9010 REM HIDEGINDITAS
9030 DIM V(50): REM VEREM
9040 DIM F$(30,7): REM FENNTARTO
TT SZAVAK
9050 FOR A=1 TO 30: READ F$(A):
NEXT A
9060 DATA "ELORE","HATRA","JOBBR
A","BALRA","ISM","ISMVEGE","","V
EGE","LE","IRD","+","-","*","/",
"ORR","ORRNE","PROGRAM","CIMEK",
"TINTA","PAPIR","HA","KOZEP","SO
S","TOLL","TOLLNE","SZIVACS","FE
LEJT","LEGYEN","KERET",""
9065 RESTORE
9070 LET PP=1: REM TOLL IGEN
9080 LET P$="": REM URESEN INDUL
9090 DIM E$(20,29): REM 20 *
ELJARAS NEVE (MAX.7 BETUSEK!) +
VALTOZOINAK SZAMA + HAROM VALTOZ
O NEVE
9092 FOR A=1 TO 20: LET E$(A,8)=
"0": NEXT A
9100 DIM E(20,4): REM A 20 ELJAR
AS CIME A P$-BAN + EGYENKENT 3
VALTOZO ERTEKE
9110 LET ELJSZAMA=1: REM AZ 1.EL
JARAS KOVETKEZIK
9120 DIM P(3): REM PARAMETEREK
9200 LET NYILTORLO=10
9210 LET VONALHUZO=20
9220 LET ORR=1
9230 LET NYILRAJZOLO=45
9240 LET FORDITO=100
9250>LET AZONOSITO=500
9260 LET VEGREHAJTO=700
9270 LET BEHATAROLO=200
9280 LET SZERKESZTO=1000
9290 LET PARANCS=1300
9300 LET ELJARAS=1700
9310 LET VALTOZO=800
9320 LET PARVALT=300: REM PARAME
TER ES VALTOZO KERESO
9500 REM MELEGINDITO
9505 GO SUB 9510: GO TO SZERKESZ
TO
9510 CLS
9520 LET X=128: LET Y=77: PLOT X
,Y: REM KEZDO KOORD.
9530 LET DG=90: REM TEKNOCIRANYI
RANY
9540 GO SUB 45: REM NYILRAJZOLO
9550 DIM R(10): REM ISM.VERME
9560 LET RSP=0: REM ISM.VEREMMUT
9570 DIM W(10): REM ISMETLESEK
SZAMLALOJA
9590 RETURN
9600 STOP
9800 SAVE "CSM-LOGO" LINE 1
```

nem ismeri. (Nincs a 9560 sorban felsoroltak között.) Ezután a B$-t feltöltjük szóközökkel, hogy meglegyen az előírt 7 betűs hossza (ha nem lenne meg), majd a feltöltött B$-t ismét „levágjuk" pontosan 7 betű hosszúságúra. Ez már beillik az E$ tömbváltozó soronkövetkező hétbetűnyi helyére. Ide azonban több nevet már nem rakhatunk, azért az ELJSZAMA mutatót megnöveljük 1-gyel. Üssük be a 1722, 1724, 1732, 1734, 1740 sorokat is. A 1750 sorral a program „megjegyezte" magának az eljárást és visszatér a SZERKESZTŐ-be.

Figyeljük meg, hogy amikor eljárást határozunk meg, a LOGO nem végzi el „magától" az eljárásban megadottakat, hanem új utasításra vár. Ha azt akarjuk, hogy végezze is el, ezután meg kell hívnunk az eljárás nevét (most már az ELJARAS nélkül!) – akkor végrehajtja az eljárást. Ehhez azonban az AZONOSITO részbe be kell építenünk az eljárások azonosítóját. (570, 580, 585)

Az 580. sor tehát végigkutatja az eljárások neveit az E$ tárolóban 1-től az ELJSZAMA-val jelölt legutóbbig bezárólag. Ha megtalálja ezek között a keresett (I$) nevet, akkor eggyel megnöveli a VM veremmutatót, és a veremnek erre az új helyére elrakja, mint a következő feladat címét, az eljárás kezdőcímét. Átállítja a V$ igen/nem típusú változóját és visszatér.

Ha nem talált eljárásnevet, az azt jelenti, hogy semmit sem tudott az illető keresett szóval azonosítani. Ilyenkor közli velünk a már korábban beírt 595. sor, hogy nem ismeri az adott szövegű parancsot, és visszatér a szerkesztőbe. Többnyire neki van igaza, és rájövünk, hogy valamit rosszul csináltunk, pl. hibásan írtunk be egy parancsot vagy meghívtunk egy eljárást, amit még nem határoztunk meg...

Ha már idáig eljutottunk, CSM-LOGOnk már „komoly" programfeladatokra is alkalmas. Tiszteljük meg tehát azzal, hogy nevet adunk neki:

```
1 REM BIT-LET CSM-LOGO
2 REM
3 REM © SZEKFU ANDRAS 1984
4 REM
5 GOTO 9000:REM – Arra az esetre, ha valaki egyszerű RUN-nal akarja elindítani a programot egy leállás után.
```

Játsszunk el egy kicsit a programmal. Mint korábban ígértük, a CSM-LOGO „rekurzív" nyelv, azaz egy eljárás meghívhatja önmagát is. Ezt már most is kipróbálhatjuk. Igaz, egyelőre az így keletkező vég nélküli („farkába harapó kígyó") programokból csak a Spectrum "BREAK" billentyűjével tudunk kilépni, de hát ez rajzolásnál nem olyan nagy bűn. Írjuk be soronként a következő eljárást:

```
ELJ SZIROM
ELORE 30
JOBBRA 163
SZIROM
VEGE
```

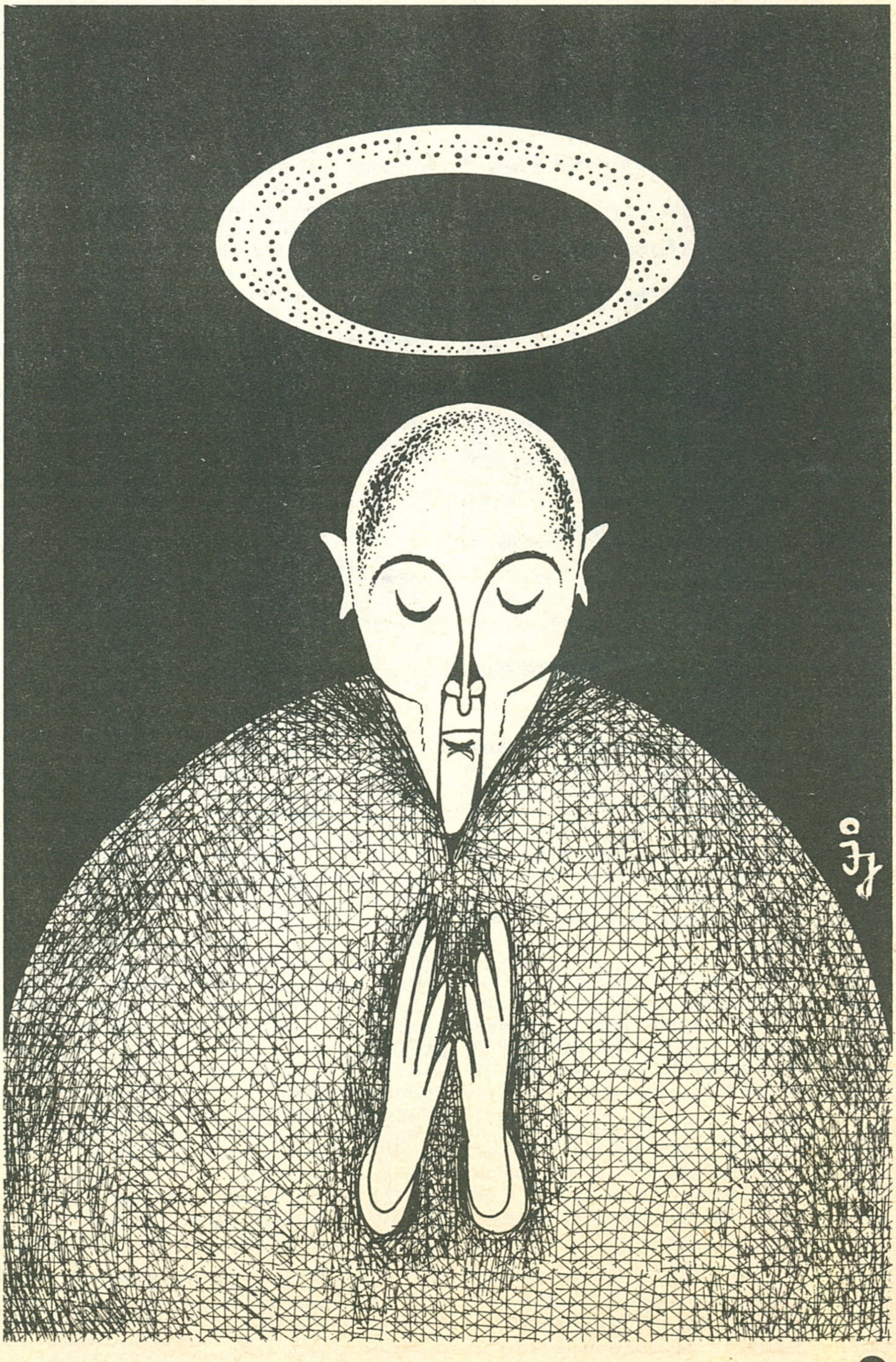
Majd hívjuk meg ezt az eljárást esetleg úgy, hogy előtte rajzoljunk egy szárat is virágunknak:

```
ELORE 40
SZIROM
```

Az ELORE 40 sor meghúzza a szárat. Utána meghívjuk a SZIROM eljárást. Ez szintén előremegy 30 teknőclépést, utána visszafordul. Azért 163 fokot választottunk, mert a 180-ból fennmaradó 17 fok csak 360 lépés után tér vissza önmagába, viszont közel áll a 15 fokhoz, amely már 24 lépés után megteszi ezt. (Tessék kipróbálni 15 fokkal is, azaz 180—15=165 fokos fordulással!) Ahogy megtörtént a 163 fokos fordulat, a program ismét önmagával találkozik: újra elkezdi a SZIROM nevű műveletsort. Elvileg így végnélkül írná egymás mellé a virágszirmokat. Gyakorlatilag ennek a verem mérete határt szab. (Hogy miért, azon ki-ki gondolkozzék kicsit saját maga! És hogy hogyan szab határt, azt tessék türelemmel kivárni!) De aki nem várja ki és unja már a szirmokat, a BREAK-kel, mint mondtuk, bármikor megállíthatja a programot. Utána, ha csak nem akarja elveszteni a SZIROM eljárást, „melegindítást" javaslunk, azaz GOTO 9500. Ha az addig rajzoltakat sem kívánja elveszteni olvasónk, akkor „még melegebb" indítással bekapcsolódhat egyenest a szerkesztő részbe: GOTO 1000.

Házi feladat kezdőknek: rajzoljon házat! Készítsen külön procedúrát (eljárást) az alapra, a tetőre, az ablakra, az ajtóra, kösse össze ezeket megfelelő teknőcmozgásokkal (emlékezzék a TOLLNE parancsra, amitől a teknőc nem húz nyomot, és a TOLL parancsra, amitől újra nyomot húz!) A sok kis eljárás végeredményeként hozza létre a HÁZ (bocsánat: HAZ) nevű szupereljárást, amely önmagában elkészíti a teljes házat!

Középhaladóknak ezentúl azt is javasoljuk, hogy bővítsék ki a CSM-LOGO utasításkészletét a leggyakoribb parancsok (pl. ELORE, JOBBRA stb.) kétbetűs változatával. Segítségül annyit, hogy a 9060-as DATA sorba kellene az új, rövidebb változatokat is elhelyezni. Ha csak ezeket kívánja, amit kevéssé javaslok, akkor lecserélheti egyszerűen a hosszúakat rövidre. Ha viszont megtartja a hosszúakat is, akkor a rövideket a DATA sor végére kell rakni. Az új parancsokat a VÉGREHAJTÓ részben a 729. utáni sorokba lehet elhelyezni. (Ha csak egy meglévőt rövidítünk, akkor az új helyre is változatlanul odaeditálhatjuk a régi sor mását.)



LOGO

ÚJ UTASÍTÁSOK: VÁLTOZÓKAT A CSM-LOGOBA!

A CSM-LOGO III. változatában 30 utasításnak van hely, azaz 30 fenntartott szónak. A 9060. sorban a szavak sorrendjét is átalakítottuk. A gyakrabban használt utasítások kerültek előbbre. Az utasítások sorrendje mindig azonos a végrehajtó blokkon (700–729) belüli programsorokéval, hiszen a végrehajtás során ennek alapján ugratjuk oda a vezérlést.

Az új utasítások a következők:

LEGYEN – a változók értékadója. LEGYEN :OLDAL 30

Érdemes rövidített alakban is beépíteni ("LE"), mert gyakran használjuk.

IRD = kiírja a változó értékét. IRD :OLDAL (kiírja:30)

+,–,*,/ – műveleti jelek. ELORE :OLDAL+5 (előtte-utána szóköz kötelező). Változók értékét pedig így módosíthatjuk: LEGYEN :OLDAL :OLDAL+2 (a BASIC-hez szokott olvasónak szokatlan, de nincs közte "=".)

PROGRAM – kiírja a LOGO-programot (P$).

CIMEK – kiírja az eljárások címeit, neveit, változóit.

TINTA – a teknőc vonalának színe. TINTA 1 (kék vonal).

PAPIR – a háttér színe.

KERET – a keret színe.

SOS – segítség oldal. Felsorolja az utasításokat.

HA – feltételes elágazás. Most még nem működik. Lásd később!

16 K-s Spectrum-tulajdonosok vigyázat! A változók bevezetése 16 K-s gépben csak úgy lehetséges, ha minden REM-et kiírtanak, és E$ (10,29)-re valamint V(20)-ra írják át a két legnagyobb tömbváltozót.

Amint a fenti példákban láthatták, a CSM-LOGO-ban a változók neve elé mindig odatapasztunk (ez esetben szóköz nélkül!) egy kettőspontot. A változó neve maga, akár a Spectrum BASIC-ben, csak betűvel kezdődhet, de tartalmazhat számot. Szóköz viszont nem lehet benne. Tehát:

:PONTSZAM
:PONT1 } – ezek mind nyelvtanilag helyesek
:PONT2A

:1.PONT } – ezek a CSM-LOGÓban hibásak.
:HIBA PONT

A 700-nál kezdődő VÉGREHAJTÓ blokkban (729-ig) azonban más változás is van. Minden olyan műveletnél, ahol valamilyen paraméter is van, beiktattunk egy GO SUB PARVALT-ot. A paraméter (tehát pl. hogy hány lépést tegyen a teknőc vagy hogy mekkorát forduljon) ugyanis mostantól kezdve nemcsak egy szám lehet, hanem egy változó is, a programnak tehát mindkettőt „fel kell ismernie".

A PARaméter-VALTozó blokk is új: 300-nál kezdődik. Itt keressük meg az utasításokat esetleg követő változók vagy paraméterek értékét. Ki kell törölni viszont a régi programból az így fölöslegessé vált 510, 520, 525, 530 és 540. sorokat.

Ha a PARVALT blokk változót talál, továbbadja a feladatot a 800-nál kezdődő VÁLTOZÓK blokknak (890-ig). A paraméterek és a változók értékeit pedig a P(3) tömbváltozóban tároljuk, amelyet deklarálnunk is illik.

(A 9120, 9310, 9320, 9330, 9340, 9350 sorok, valamint a PROGRAM, CIMEK és az ERTEKADAS blokkok a listán olvashatók.)

ÚJ LEHETŐSÉGEK: ÉLJEN A SPIRÁL!

Mi újat tud ezekután a CSM-LOGO? Írjuk be például a következő LOGO-parancsot:

```
ELJARAS SPIRAL :OLDAL :SZOG :UJRA
  ORRNE
  ISM :UJRA
    ELORE :OLDAL
    JOBBRA :SZOG
    LEGYEN :OLDAL :OLDAL + 2
  ISMVEGE
VEGE
ELJARAS VALTOZOK :OLDAL :SZOG
:UJRA
  LEGYEN :OLDAL 3
  LEGYEN :SZOG 90
  LEGYEN :UJRA 20
VEGE
```

Először hívjuk meg a VALTOZOK eljárást. Látni fogjuk, hogy a változók tényleg megkapták a kívánt értékeket. Utána hívjuk meg a SPIRAL eljárást is! Az 1. ábrán látható rajzot kapjuk. SZIVACS-csal töröljük le, KÖZEP-pel vigyük a teknőcöt középre, LEGYEN :SZOG 117-tel adjunk új értéket a :SZOG változónak, és SPIRAL-lal indítsuk el újra a rajzolást! (lásd az ábrát).

Ameddig nincsen még IF-THEN (HA-AKKOR) típusú feltételutasításunk, addig az ismétlések számának (:UJRA változó) szabályozásával dönthetjük el, hogy SPIRALt akarunk. Ne feledjék el, hogy a LOGO (a PASCAL-hoz hasonlóan) szigorúbban kezeli a változókat mint a BASIC. Minden változót valamely ELJÁRÁS első sorában deklarálnunk kell, különben a program nem fogadja el a LEGYEN értékadó utasítást, hanem közli, hogy A... VALTOZO NINCS DEKLarálva. BASIC-hez szokott reflexeinknek ez eleinte szokatlan lesz, de programozási stílusunk okvetlen javulni fog a ránk kényszerített rendszeretettől. Ezért is mondják, hogy a LOGO gyerek és felnőtt számára a komolyabb programozás (és a PASCAL, a PROLOG, a LISP) előszobája.

A FELTÉTELES ELÁGAZÁS: HA

A BIT-LET-ben közölt CSM-LOGO még nem tartalmazott feltételes elágazást, azaz a BASIC IF-THEN-jéhez hasonló utasítást. Nem véletlenül, ugyanis a CSM-LOGO program belső szerkezete (főleg az, hogy a LOGO-programot egyetlen, dimenzionálatlan stringváltozóban tároljuk – P$ –) kissé megnehezíti az elágazás szellemes megoldását. Kísérletképpen egy kis szépséghibával azonban megkerültük a nehézségeket és az itt közölt 7. sz. CSM-LOGO változatban már működik a feltételes elágazás. A kis szépséghiba annyi, hogy a HA utasítás nyelvtana a következő:

HA :OLDAL = 3 VEGE) – ahol a zárójel (SYMBOL SHIFT 9) kötelező a sor végén.

Hát ez bizony nem valami elegáns megoldás. El is tér a LOGO-nyelv „profi"-változataitól. Azonban a CSM-LOGO szerkezeti logikáján belül maradva, jobb egyelőre nem jutott eszünkbe. Ismét mondjuk: a CSM jelentése – csináld magad! Tehát kíváncsian várjuk hogy az olvasók közül eszébe jut-e valakinek elegánsabb megoldás a feltételes elágazásra. Addig is lássuk, hogyan működik ez.

Ha az AZONOSITO "HA" utasítást talált, a 721. sorba küldi a vezérlést. Innen a PARVALT eljárásra ugrik a program, hogy átvegye a HA utáni változó vagy szám értékét.

A 2300–2350-es sorok kiemelik a HA utasítás után következő relációjelet (az = jelen kívül használható a <, >, <=,<=, <> is).

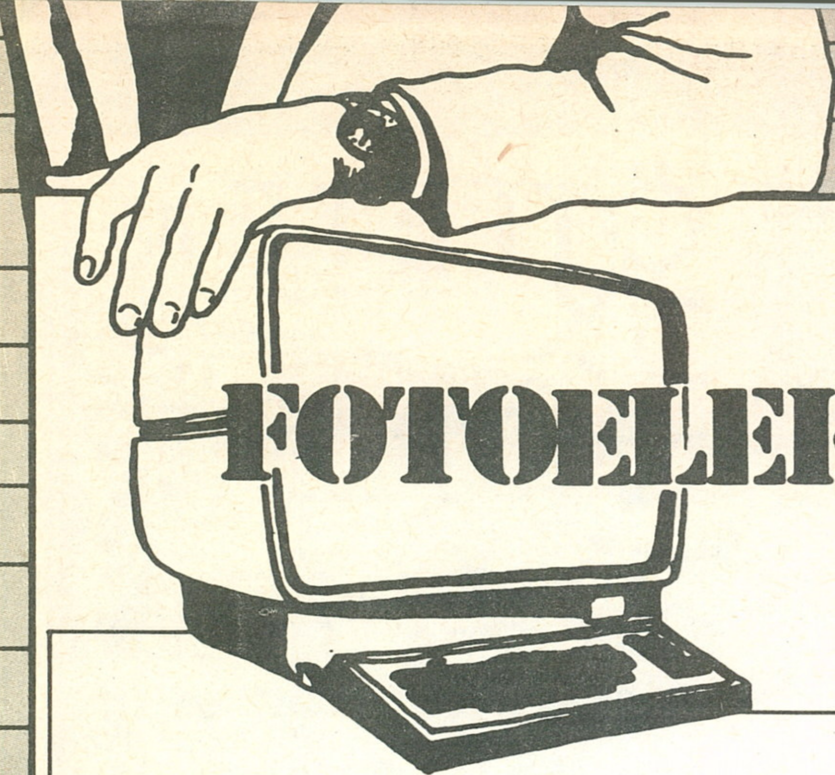
Utána a paraméter sorszámát 2-re módosítjuk, és ezúttal a PARVALT rutin közepébe ugrunk (330), hogy kiderüljön, mihez is hasonlítja a relációjel az első paraméter értékét. Ezzel tehát feltételezett reláció. Tehát ha a fentebb említett példásor futásakor az :OLDAL változó éppen a 4 értéket vette fel, akkor itt már az :OLDAL = 3 rész a változó konkrét értékével szerepel, tehát így: 4=3.

Tudjuk, a feltételes elágazás lényege éppen az, hogy az itt következő műveletet a LOGO csak akkor hajtja végre, ha a fenti egyenlőség IGAZ. A végrehajtás–nem végrehajtás a CSM-LOGO-ban azt jelenti, hogy HAMIS esetben a program a rákövetkező utasítással folytatja, tehát kihagyja a HA utáni első utasítást. És ezért kell a HA sor végére egy külön jel : innen tudja meg a program, hogy meddig kell ugornia, ha a feltétel nem igaz, hanem HAMIS. Ezért a 2400–2440 sorok megkeresik a HA sor végén a zárójelet, és ennek címét elteszik a ZCIM nevű BASIC változóba. (Ha nem találnak zárójelet, akkor hibajelentést írnak.) Ha megvan a zárójel címe, következik a tulajdonképpeni döntés, a „HA-művelet". (2450–2470) A 2460. sor a példánkban szereplő 4=3 relációt egyetlen stringgé alakítja, majd a Sinclair BASIC adta lehetőséggel élve, a VAL utasítással kiértékeli. Ha a reláció hamis, mint példánkban, akkor ez az érték 0 lesz, és ilyenkor a LOGO program végrehajtási sorrendjét átirányítják a zárójel után következő (tehát már nem feltételes) újabb utasításra. Ha viszont a reláció igaz volt, akkor végrehajtódik a zárójel ELŐTTI utasítás. Ha a feltétellel bonyolultabb feladatot vezérlünk, ajánlatos ebből egy külön ELJÁRÁST írni, és azt hívni meg a HA-val kezdődő sorban.

Sajnos a 16 K-s Spectrummal rendelkező olvasóinktól itt már búcsút kellett vennünk. Ha netán sikerül is az összes REM elhagyásával és egyéb trükkökkel belezsúfolni a 9 K-ba (hiszen a képernyőn kívül ennyi marad a 16-ból) a CSM-LOGO 7. változatát, igazából nem éri meg, mert nem marad hely a P$-nak azaz maguknak a LOGO-programoknak. A 48 K-s Spectrum-tulajdonosok persze még fejleszthetik a programot, például hibajavítási rutinokkal (hogy ne kelljen a hibás részeket elejétől újra írni) vagy a LOGO-programok külön kimentésének lehetőségével. (Ez utóbbihoz tudni kell, hogy a Spectrum csak dimenzionált stringet tud a DATA utasítással kimenteni, tehát a mi P$-ünket nem. Kimentés előtt tehát – ha van még helyünk – dimenzionálni kell egy LEN P$ hosszúságú másik stringváltozót és azt lehet kimenteni. De ki kell mentenünk az eljárások neveit és címeit is.)

Se az említett fejlesztési ötletekkel, se magával a CSM-LOGO különböző változataival nem azt céloztuk meg, hogy a nagy szoftvercégek gépi kódban írott profiprogramjaival konkuráljunk. A csináld magad, az együttgondolkozás öröme volt a célunk. Ezért itt külön is megköszöni a szerző azoknak a BIT-LET-olvasóknak érdeklődését, akik segítségükkel, javaslataikkal, jogos bírálataikkal az első közlések idején megkeresték vagy akik a CSM-LOGO alapján más gépekre hasonló programokat írtak. Azoknak viszont, akik nem kívánnak ennyi programsort beírni gépükbe, de mégis meg akarják szerezni a CSM-LOGO-t maguknak vagy gyermekeiknek, változatlanul rendelkezésre áll az INTEGRAL Gm (1013 Bp., Lánchíd u. 15. fsz. 2. tel.: 360-406), ahol a CSM-LOGO a 48 K-s Spectrumra, illetve egy szerényebb, kevesebbet tudó változat a 16 K-s ZX 81-re, 279 Ft-ért utánvéttel megrendelhető, (kérjük a géptípust megjelölni!) vagy telefonos megbeszélés alapján személyesen átvehető.

Szekfű András



FOTOELEKTRONIK ISZ

Szövetkezetünk SZEMÉLYI SZÁMÍTÓGÉPEK, Hi-Fi, videó, fotó és egyéb műszaki cikkek adás-vételével foglalkozik!

KÍNÁLATUNKBÓL:

Commodore 16, 116, 64, 128, +4, 720, PC 10, PC 20
Spectrum 48 K, Spectrum +, Sinclair QL számítógépek,
Commodore SFD 1001 1 MB, Commodore, 8250 LP,
IBM PC XT, IBM PC AT konfigurációk,
irodatechnikai készülékek.

Egyes készülékekre garanciát biztositunk!
A megrendeléseket rövid átfutási idővel teljesítjük!
Igény esetén lízing szervezését is vállaljuk!
Szaküzleteinket Budapesten és vidéken egyaránt megtalálja!

FOTO-ELEKTRONIK
IPARI SZÖVETKEZET

SZAKÜZLETEINK:

Budapesten: I., Fő u. 37/c. 159-869 • **V.,** Magyar u. 1. 178-854 • **V.,** Múzeum krt. 19. 173-043 • **V.,** Váci u. 23. 183-240 • **VI.,** Szív u. 38. • **VII.,** Akácfa u. 59. 222-278, 217-131 • **VIII.,** Baross tér 6. 134-116 • **VIII.,** József krt. 40. 131-478 • **IX.,** Ráday u. 9. 176-093 • **XI.,** Móricz Zs. körtér 7. 868-787 • **XIII.,** Rajk L. u. 46/b 299-604 • **XIV.,** SUGÁR – Örs vezér tér 836-567 •

Vidéken: Debrecen, Szabó I. altb. tér 6. 52-12-068 • **Eger,** Széchenyi u. 5. • **Győr,** Bem tér 1. 96-12-802 • **Kaposvár,** Füredi u. 24. 82-15-351 • **Miskolc,** Korvin O. u. 5. 46-38-965 • **Pápa,** Fő tér 14. 89-24-402 • **Pécs,** Jókai u. 5. 72-14-302 • **Székesfehérvár,** Széchenyi u. 15/a 22-18-228 • **Szombathely,** Tolbuhin u. 33. 94-18-277 •

ÖTLETEK PROGRAMOK



FILMCSINÁLÓ

A digitális technika egyre újabb és újabb területeket hódít meg. Ilyen például a filmgyártás. Olvashattuk a sajtóban, hogy az Egyesült Államokban elkészült az első számítógép segítségével készített, egész estét betöltő film, a TRON. Láthattunk a televízió egyik közelmúltbeli HÉT c. műsorában részleteket egy számítógépes animációs filmből. Ebből azt is megtudhattuk, hogy a berendezés, melyen az ilyen film előállítható, méregdrága és igen bonyolult.
Így hát jogos kétkedéssel csóválhatja az olvasó a fejét a cikk címének olvastakor, hogy jön ide a ZX 81? Pedig nincs tévedés! Az alábbiakban részletesen ismertetünk egy mozgófilm-előállító programot Sinclair ZX 81-re, mellyel bárki elkészítheti saját rajzfilmjét. Természetesen a grafika egyszerű lesz, amilyen egyszerű a ZX 81 állóképes félgrafikája, a színek világától továbbra is el leszünk zárva, és a „film" hossza is korlátozott (legfeljebb 62 képkocka), de ezeken a korlátokon belül a mozgás teljesen szabad lesz, érdemes a programot kipróbálni!
Mielőtt elmondanánk, hogy is kell a programot használni, nézzük meg, hogyan kell a számítógépbe betölteni.
A program alapvetően BASIC-ben íródott, három rövid gépi kódú rutin felhasználásával. A gépi kódú rutinokat a program 1. sorának REM utasítása tartalmazza. A program betöltését célszerű ennek a sornak a beírásával kezdeni. A betöltést az 1. listán látható segédprogram segítségével végezhetjük el legegyszerűbben. Ennek első sora egy 73 db. számjegyet tartalmazó REM utasítás.
Ha a programot RUN-nal elindítjuk, a 2. lista kódjait bePOKE-olhatjuk a REM-be. Ha jól végeztük az adatok megadását, a képernyőn visszakapjuk a 2. listát (baloldalt a RAM címét, jobboldalt a kódot tartalmazza).
A program ellenőrzésképp összeadja a bevitt adatokat, és ha jól dolgoztunk, az ellenőrzési összegnek 7650-nek kell lennie.
Ha most a töltőprogramot újra kilistázzuk, azt tapasztaltjuk, hogy 1. sora megváltozott, mégpedig a 3. lista 1. sorával fog megegyezni. Ez is volt a célunk, előállítottuk a tulajdonképpeni programunk végleges 1 sorát a gépi kódú szubrutinokkal.
Ezután a segédprogram 20–180 sorszámú sorát kitöröljük és beírjuk a főprogram (3. lista) többi BASIC utasítását. Ha készen vagyunk a program betöltésével, célszerű átmásolni kazettára. A másolást GOTO 3-mal vagy RUN-nal indítjuk. Így a visszaolvasás során a program a betöltés után automatikusan elindul.
A program beolvasása a számítógép gépkönyve szerint történik. Legalább 16 K memória szükséges a használatához, 16 K memória esetén a gép bekapcsolása után LOAD „MOZI"-t billentyűzünk, majd a magnetofon elindítása után „NEW LINE"-t nyomunk meg. Ha a gép memóriája 16 K-nál nagyobb, úgy előbb a RAMTOP rendszerváltozót kell elállítani az alábbi utasításokkal, pl. 64 K memória esetén:
POKE 16388, 255
POKE 16389, 255
NEW
LOAD "MOZI"
A program betöltés után automatikusan elindul, és felírja a címet, valamint az adott memóriánál szerkeszthető leghosszabb film képeinek számát. (16 K esetén 14 kép, 64 K esetén 62 kép). Ha nagyobb memória esetén is csak 14 képet ír ki a program, akkor elfelejtettük a RAMTOP-ot elállítani. Ilyenkor a műveleteket elölről kell kezdeni.
Ha ismét megnyomunk egy gombot, a képernyőt a program kitörli, és a kép kialszik néhány másodpercre. Miután kivilágosodik, az egész kép üres lesz, csak a bal alsó sarokban fog egy inverz (fekete alapon fehér) „U" kurzor villogni. A program most működésben van. A kurzor, nemcsak U, hanem K, A, G is lehet, ezek különböző üzemmódokat jelölnek, amelyet azért fontos tudni, mert az egyes üzemmódokban az egyes billentyűk funkciója más. Az alábbiakban ezeket ismertetjük:

[U] – átlátszó, „üveg" üzemmód
A kurzort a nyilakkal mozgatva a képernyő eredeti tartalmát nem változtatja meg. Olyan, mintha a kurzor egy különálló kocka volna, és ezt az ablakon végighúznánk, a mögötte levő képet csak pillanatnyilag takarja.
A billentyűk funkciója:
- NEW LINE átvált [K] üzemmódba
- SHIFT 5 (←): a kurzort balra mozgatja
- SHIFT 6 (↓): a kurzort lefelé mozgatja
- SHIFT 7 (↑): a kurzort felfelé mozgatja
- SHIFT 8 (→): a kurzort jobbra mozgatja
- SHIFT 9 (GRAPHICS): átvált [G] üzemmódba
- SHIFT NEWLINE (FUNCTION): átvált [A] üzemmódba
- BREAK: megszakítja a program futását.

[G] – grafikus üzemmód
Ebben az üzemmódban a nyilakkal jelölt gombok hatástalanok, csak azt lehet vezérelni, hogy milyen grafikus karakter legyen a villogó kurzor helyén a képernyőn.
- SHIFT 1–8:
- SHIFT Q–Y:
- SHIFT A–H:

} A megfelelő félgrafikus jelre vált a kurzor alatt

- 0: üres karaktert (SPACE-t) ír a kurzor alá
- SHIFT 0: fekete négyzetet ír a kurzor alá
- U: átvált [U] üzemmódba
- NEWLINE: átvált [K] üzemmódba
- SHIFT NEW LINE (FUNCTION): átvált [A] üzemmódba
- BREAK: leállítja a program futását.

[A] – alfanumerikus üzemmód
A nyilak hatástalanok itt is. – A billentyűk feketével jelölt (SHIFT nélküli) karakterét nyomtatja a képernyő soronkövetkező helyére (jobbra, ill. a sor végén a következő sor első karakterébe). Vigyázni kell, mert kb. 1 s múlva a lenyomott billentyű ismétel!
- SHIFT 9: átvált [G] üzemmódba
- NEW LINE: átvált [K] üzemmódba
- BREAK: megszakítja a program futását.

[K] – rajzoló üzemmód
- SHIFT 5–8: A megfelelő nyíl irányába mozog a kurzor, és az új helyre írja a karaktert, „otthagyja a nyomát". Ha a beállított karakter SPACE (üres karakter), a kurzor mozgatása a radírozásnak felel meg.
- SHIFT 9: átvált [G] üzemmódba
- SHIFT NEW LINE (FUNCTION): átvált [A] üzemmódba
- U: átvált [U] üzemmódba
- V: törli a teljes képernyőt („CLS")
- N: a képet inverzére változtatja, vagyis a fehér alapon fekete karakterből fekete alapon fehér karaktereket csinál
- S: a képernyőn látható képet átmásolja a kívánt filmkockába („SAVE")
- J: a megadott filmkockát visszamásolja a képernyőre („LOAD")
- R: a megadott számú képkockából álló filmet leforgatja anynyiszor egymás után, ahogy a felhasználó kívánja („RUN")
- Z: átmenti a filmet szalagra.

A szalagramentés előtt a program kéri a film FILE-nevét, majd megáll, és a képernyőn megjelenő felirat utasítása szerint kell a magnetofont elkészíteni és a programot újraindítani.

1. lista

```
   1 REM 12345678901234567890123
4567890123456789012345678901234
5678901234567890123
  20 DIM A$(73)
  30 LET A=16514
  40 LET S=0
  50 FOR I=1 TO 73
  60 INPUT B
  70 PRINT A;TAB 10;B
  80 LET S=S+B
  90 LET A$(I)=CHR$ B
 100 LET A=A+1
 110 NEXT I
 120 PRINT "ELLENORZESI OSSZEG:
";
 130 PRINT S
 140 LET A=16514
 150 FOR I=1 TO 73
 160 POKE A,CODE A$(I)
 170 LET A=A+1
 180 NEXT I
```

Ha valami hiba történne a másolásban, és ezt észrevesszük közben, a magnót le kell állítani, visszacsévélni, a programot pedig vagy a BREAK gombbal leállítani és a képernyő előző utasítása szerint újraindítani, vagy megvárni, míg megjelenik az [U] kurzor.

Ekkor az ismertetett módon a program segítségével jussunk el a másolásig: ([U] → [K] → [Z])

Ha a filmet vissza akarjuk játszani, akkor az új FILE-névvel kell betölteni a programot a számítógépbe.

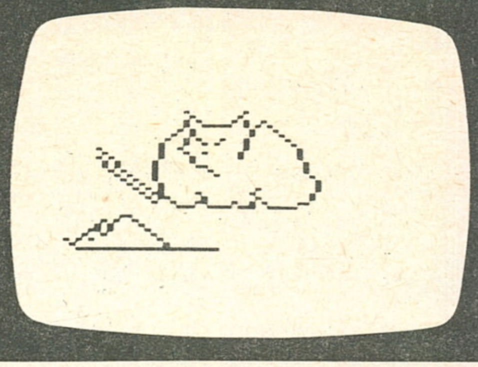
Fontos figyelmeztetés: **SOHA NE HASZNÁLJON RUN VAGY CLEAR PARANCSOT,** mert a program nem fog működni, és az elkészült film megsemmisül!

A fentieken kívül a program mindig kiírja, hogy milyen adatot vár, vagy mit kell tennie a kezelőnek.

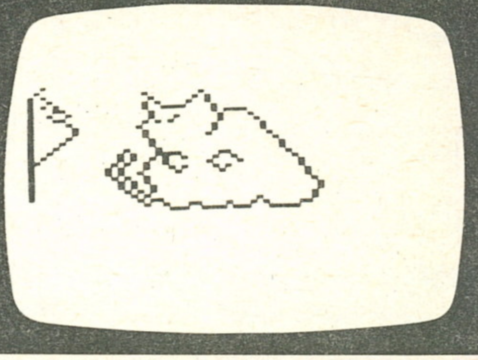
A program működését célszerű példákon begyakorolni, és csak utána hozzákezdeni egy hosszabb munkához. A gyakorlat sok hasznos ötletre vezeti rá a felhasználót.

Hogyan működik a program?
Az alapelv igen egyszerű. Az eddigiekből kiderült, hogy nagy része egy egyszerű állóképszerkesztő program, amelyhez hasonlót csaknem minden új számítógép-tulajdonos ír magának.

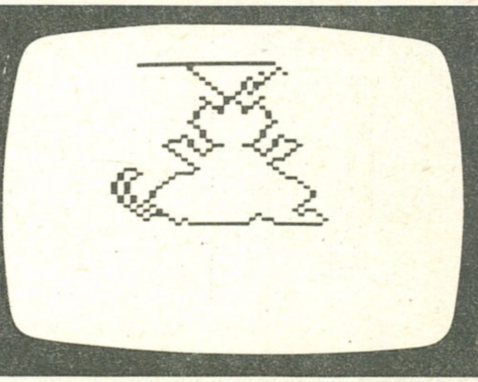
Ebből úgy lesz mozifilm, hogy az egymás után következő fázisképeket állóképként megszerkesztjük, majd a képeket



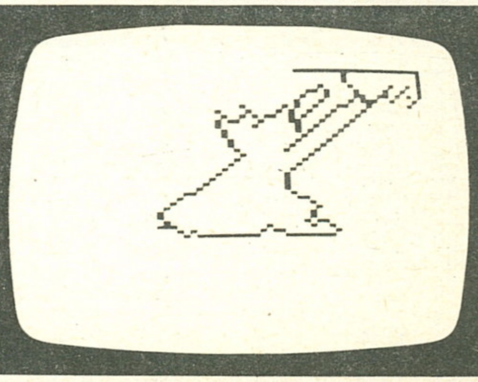
1



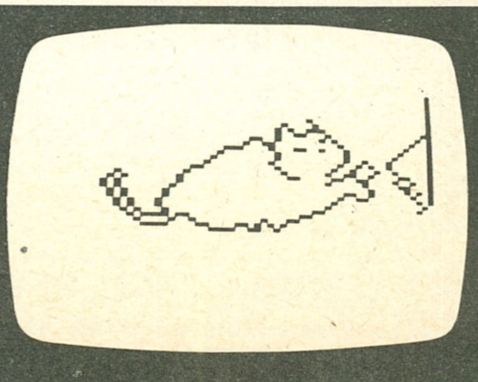
2



3



4



5

3. lista

```
   1 REM E£RND7:+▗▘4▖LEN █/▝CHR$
█NOT AT LPRINT 7( GOSUB 7$? RETU
RN  4 SLOW TAN GOSUB ??RNDE£RND:
P▗▘▝ GOSUB K< GOSUB ??RNDTAN E?RN
D:P▗▘▝ GOSUB ?£RND GOSUB K  GOSUB
 ??RNDTAN TAN
   3 SAVE "MOZI"
   4 GO TO 100
   5 IF INKEY$="" THEN GO TO 5
   6 CLS
   7 RETURN
  40 LET X$(I,1)=X$(I,1)
  41 POKE 16507,PEEK 16402
  42 POKE 16508,PEEK 16403
  43 RETURN
  50 LET D=PEEK 16507
  51 LET E=PEEK 16508
  52 LET DD=16507
  53 LET EE=16508
  54 RETURN
 100 PRINT AT 3,7;"MOZGOKEP SZER
KESZTO"
 110 PRINT "          -------------
------"
 120 PRINT ,,TAB 13;"PROGRAM"
 130 PRINT TAB 13;"-------"
 140 PRINT AT 10,9;"KESZITETTE,
CO.:"
 150 PRINT AT 13,10;"KABOLDY PET
ER"
 160 PRINT AT 18,12;"* 1984 *"
 170 LET Q=-1
 180 LET B$=" "
 190 DIM X$(1)
 200 GO SUB 3000
 220 LET NKEP=INT ((RAMTOP-100-E
LINE)/21/33)
 230 PRINT AT 20,2;"A KESZITHETO
 KEPEK SZAMA: ";NKEP
 240 GO SUB 5: REM
 260 DIM X$(NKEP,21*33)
 270 FOR J=1 TO NKEP
 280 PRINT AT 1,0;J;". KEPKOCKA"
 290 GO SUB 2560: REM
 310 NEXT J
 320 CLS
 330 LET Z$="
"
 340 SLOW
1010 LET X=0
1020 LET Y=21
1040 LET K$="U"
1050 PRINT AT Y,X;K$
1051 LET NN=SIN PI
1060 LET C$=INKEY$
1070 PRINT AT Y,X;B$
1080 LET C$=INKEY$
1090 IF C$="" THEN GO TO 1130
1100 LET A=CODE C$
1110 GO SUB (Q*1000+5000+3*A)
1120 GO TO 1050
1130 LET NN=SIN PI
1140 GO TO 1050
1200 REM COPY
1210 CLS
1220 PRINT AT 5,4;"A KESZ FILMET
 ATMASOLJUK",TAB 12;"KAZETTARA"
1230 PRINT AT 10,4;"A FILM FILE-
NEVE:  ";
1240 INPUT T$
1250 PRINT T$
1260 PRINT AT 18,0;"INDITSA EL A
MAGNOT RECORD UZEM-      MODBAN E
S BILLENTYUZZON           ""GOTO
1270""-ET, MAJD ""NEWLINE""-T"
1265 STOP
1270 SAVE T$
1280 GO SUB 3570
1290 LET Q=-1
1310 GO TO 1000
1700 REM U
1720 LET K$="U"
1730 LET Q=-1
1740 RETURN
1800 REM A
1810 LET B$=C$
1820 IF X<=30 THEN GO TO 1900
1830 IF Y<=20 THEN GO TO 1860
1850 RETURN
1860 LET Y=Y+1
1870 LET X=0
1890 RETURN
1900 LET X=X+1
1940 RETURN
2000 REM LOAD
2010 PRINT AT 21,0;Z$;AT 21,1;"H
ANYADIK KOCKAT HOZZA AT?"
2110 GO SUB 2500
2120 RAND USR 16568
2130 PRINT AT 21,0;Z$;AT 21,1;J;
". KOCKAT MASOLTUK AT"
2490 RETURN
2500 REM CIM
2550 INPUT J
2560 IF J<1 OR J>NKEP THEN GO TO
 2550
2570 LET I=INT J
2590 GO SUB 40
2600 RETURN
2700 REM SAVE
2710 PRINT AT 21,0;Z$;AT 21,1;"H
ANYADIK KOCKABA MASOLJA?";
2770 GO SUB 2500
2780 RAND USR 16549
2800 PRINT AT 21,0;Z$;AT 21,1;J;
". KOCKABA MASOLTA";
2990 RETURN
3000 LET DFILE=PEEK 16396+256*PE
EK 16397
3010 LET VARS=PEEK 16400+256*PEE
K 16401
3020 LET ELINE=PEEK 16404+256*PE
EK 16405
3030 LET RAMTOP=PEEK 16388+256*P
EEK 16389
3040 RETURN
3500 REM RUN
3510 PRINT AT 21,0;Z$;AT 21,1;"H
ANY KOCKABOL ALLJON A FILM?";
3520 INPUT II
3530 IF II<1 OR II>NKEP THEN GO
TO 3520
3540 PRINT AT 21,0;Z$;AT 21,1;"H
ANYSZOR JATSSZUK LE A FILMET?";
3550 INPUT J
3560 IF J<1 OR J>100 THEN GO TO
3550
3570 LET II=INT II
3580 LET I=1
3582 GO SUB 40
3584 GO SUB 50
3590 FOR K=1 TO J
3600 POKE DD,D
3605 POKE EE,E
3610 FOR L=1 TO II
3620 RAND USR 16568
3630 NEXT L
3640 NEXT K
4335 RETURN
4336 GO SUB 7336
4337 LET B$=CHR$ PEEK (DFILE+Y*3
3+X+1)
4338 RETURN
4339 GO SUB 7339
4340 LET B$=CHR$ PEEK (DFILE+Y*3
3+X+1)
4341 RETURN
4342 GO SUB 7342
4343 LET B$=CHR$ PEEK (DFILE+Y*3
3+X+1)
4344 RETURN
4345 GO SUB 7345
4346 LET B$=CHR$ PEEK (DFILE+Y*3
3+X+1)
4347 RETURN
4348 LET K$="G"
4349 LET Q=1
4353 RETURN
4354 LET K$="K"
4355 LET Q=2
4356 RETURN
4362 RETURN
4363 LET K$="A"
4364 LET Q=0
4365 RETURN
5189 RETURN
5190 GO SUB 1800
5191 RETURN
5347 RETURN
5348 LET K$="G"
5349 LET Q=1
5350 RETURN
5353 RETURN
5354 LET K$="K"
5355 LET Q=2
5356 RETURN
5357 LET B$=" "
6083 RETURN
6084 LET B$=" "
6173 RETURN
6174 REM U
6175 GO SUB 1700
6335 RETURN
6336 LET B$="▀"
6337 RETURN
6339 LET B$="▄"
6340 RETURN
6342 LET B$="▌"
6343 RETURN
6345 LET B$="▐"
6350 RETURN
6351 LET B$="▘"
6352 RETURN
6354 LET K$="K"
6355 LET Q=2
6356 RETURN
6357 LET B$="█"
6362 RETURN
6363 LET K$="A"
6364 LET Q=0
6575 RETURN
6576 LET B$="▟"
6647 RETURN
6648 LET B$="▒"
6649 RETURN
6651 LET B$="▙"
6652 RETURN
6654 LET B$="▝"
6655 RETURN
6657 LET B$="▜"
6658 RETURN
6660 LET B$="▚"
6661 RETURN
6663 LET B$="▞"
6664 RETURN
6666 LET B$="▗"
6667 RETURN
6669 LET B$="▖"
6670 RETURN
6672 LET B$="▛"
6673 RETURN
6675 LET B$="▒"
6676 RETURN
6678 LET B$="▒"
6679 RETURN
6681 LET B$="▒"
6682 RETURN
6684 LET B$="▒"
6685 RETURN
6687 LET B$="▒"
7140 RETURN
7141 REM LOAD
7142 GO TO 2000
7152 RETURN
7153 PRINT AT 0,0;
7154 RAND USR 16514
7164 RETURN
7165 REM RUN
7166 GO TO 3500
7167 RETURN
7168 REM SAVE
7169 GO TO 2700
7173 RETURN
7174 REM UVEG
7175 GO SUB 1700
7176 RETURN
7177 REM CLS
7178 CLS
7188 RETURN
7189 REM COPY
7190 GO TO 1200
7335 RETURN
7336 IF Y=0 THEN RETURN
7337 LET Y=Y-1
7338 RETURN
7339 IF Y=20 THEN RETURN
7340 LET Y=Y+1
7341 RETURN
7342 IF X=0 THEN RETURN
7343 LET X=X-1
7344 RETURN
7345 IF X=31 THEN RETURN
7346 LET X=X+1
7347 RETURN
7348 LET K$="G"
7349 LET Q=1
7362 RETURN
7363 LET K$="A"
7364 LET Q=0
7997 RETURN
8130 LET Q=-1
8140 LET K$="U"
8150 GO TO 1050
```

2. lista

16514	42	16527	48	16540	237	16553	42	16566	64	16579	237
16515	12	16528	4	16541	35	16554	12	16567	201	16580	176
16516	64	16529	198	16542	13	16555	64	16568	42	16581	0
16517	35	16530	128	16543	121	16556	14	16569	123	16582	237
16518	14	16531	24	16544	254	16557	181	16570	64	16583	99
16519	21	16532	2	16545	0	16558	6	16571	14	16584	123
16520	6	16533	214	16546	32	16559	2	16572	181	16585	64
16521	32	16534	128	16547	228	16560	237	16573	6	16586	201
16522	126	16535	215	16548	201	16561	176	16574	2	ELLENŐRZÉSI	
16523	229	16536	193	16549	237	16562	19	16575	237	ÖSSZEG:	7650
16524	197	16537	225	16550	91	16563	237	16576	91		
16525	254	16538	35	16551	123	16564	83	16577	12		
16526	118	16539	16	16552	64	16565	123	16578	64		

4. lista

```
16514  30      ORG   16514
16514  40      LD    HL,(16396)
16517  50      INC   HL
16518  60      LD    C,22
16520  70  C1  LD    B,32
16522  80  C2  LD    A,(HL)
16523  90      PUSH  HL
16524 100      PUSH  BC
16525 110      CP    118
16527 120      JR    NC,C3
16529 130      ADD   A,128
16531 140      JR    C4
16533 150  C3  SUB   128
16535 160  C4  RST   16
16536 170      POP   BC
16537 180      POP   HL
16538 190      INC   HL
16539 200      DJNZ  C2
16541 210      INC   HL
16542 220      DEC   C
16543 230      LD    A,C
16544 240      CP    0
16546 250      JR    NZ,C1
16548 260      RET
16549 270  C5  LD    DE,(16507)
16553 280      LD    HL,(16396)
16556 290      LD    C,25
16558 300      LD    B,3
16560 310      LDIR
16562 320      INC   DE
16563 330      LD    (16507),DE
16567 340      RET
16568 350  C6  LD    HL,(16507)
16571 360      LD    C,25
16573 370      LD    B,3
16575 380      LD    DE,(16396)
16579 390      LDIR
16581 400      NOP
16582 410      LD    (16507),HL
16585 420      RET
16586 430      END
```

szép sorban a program X$ változójába lerakjuk. A film visszajátszásakor az ilyen módon előre elkészített „képkockákat" másoljuk vissza gyors egymásutánban a képernyőre.
A ZX 81-nél a képernyőn látható képet a RAM memória Display File nevű része írja el. Ez a memóriaterület a D-FILE rendszerváltozóban tárolt címmel kezdődik, és a képernyőn pillanatnyilag látható karakterek kódját tartalmazza sorfolytonosan, minden sor végén egy NEW LINE-karaktert. Így a Display-File hossza 21x33 = 693 byte.
A Display-File helye a ZX 81-ben kötött. A változók helye azonban sok mindentől függ egy programon belül is. (Megváltozik, ha törlünk vagy beírunk, vagy új BASIC-sort, vagy változót hívtunk be stb.) Ezért mielőtt az X$ változóba a képet be- vagy kiírnánk, a program megkeresi az X$ stringtömb helyét a memóriában. Ezt a műveletet a 40-es sornál kezdődő szubrutin hajtja végre. A rendszerváltozókat pedig a 3000-es sornál kezdődő szubrutin kérdezi le (ez utóbbi más programokban is jól használható!).
A látható és a háttérben (X$-ben) levő kép helyének meghatározása után az átmásolást két rövid gépi kódú szubrutin végzi el a Z80 assembly nyelv LDIR utasítása segítségével. A harmadik gépi kódú szubrutin a teljes kép inverzre fordítását végzi el. A nyomtatható karakterek kódjai a ZX81-ben 0–63 közé esnek, az inverz karaktereké 128-cal több, mint a normál karaktereké, és kódjuk 128 és 191 közé esik. A rutin ezt a szabályt használja ki. A képernyőn megvizsgál minden karakterkódot és ha 128-nál kisebb, 128-at ad hozzá, ha 128-nál nagyobb, 128-at levon belőle.
A BASIC program eléggé rugalmas ahhoz, hogy teret hagyjon az olvasó további ötleteinek megvalósításához. Így például könnyen megoldható sprite-ok definiálása és mozgatása álló háttér előtt, több sprite, ill. szabadon mozgó figura mozgatása egymás fölötti síkokban stb., stb.
A kész „film" lejátszásához nincs szükség a program képszerkesztő rutinjaira, így az elkészítés után ezek a programrészek törölhetők, és új BASIC sorokkal helyettesíthetők. Csak arra kell ügyelnünk, hogy RUN vagy CLEAR utasítást ne adjunk ki, és ne dimenzionáljuk át az X$ string változót, mert akkor a film megsemmisül.
Végül azok számára, akik járatosak az assembler nyelvben, közöljük a gépi kódú rutinok assembler-listáját (lásd: 4. lista).
Jó szórakozást!

dr. Kaboldy Péter

ZENÉS ZX

A program segítségével házi számítógépünket kis szintetizátorként használjuk. A gép megszólaltatásához a „MIC" kimenetjelét kell felerősíteni.
A program gépi kódú, és BASIC részekből áll. Elkészítéséhez felhasználtam egy ZX 81-re írt hanggeneráló szubrutint. A gépi kódú rész betöltése a következő oldal alján lévő „Beszélő ZX" című program 1–6. sorának segítségével történik oly módon, hogy az első REM sorba összesen 401 darab „Z" karaktert, a program 2-es sorában pedig 16593 helyére 16658-at írunk. Ezt a programot futtatva írjuk be az 1. számú táblázat hexadecimális számait.
Ezután ezt a programot az 1-es sor kivételével töröljük ki, illetve egyidejűleg gépeljük be a következő oldalon levő főprogramot. Ha ezzel elkészültünk, akkor adjuk ki a RUN 500 parancsot. Az 1-es gomb lenyomásával megkezdhetjük a II. számú táblázat alapján a hangolást. A számpárok beírása után írjunk két 0-t, ezzel a program visszatér a választási lehetőségekre. Az 5-ös lenyomásával tudjuk a szokásos módon rögzíteni a programot. A billentyűzet alsó két sora zongoraként használható a szintetizátor üzemmódban. Így a gép két oktávot fog át a

I. táblázat

CD	BB	02	44	4D	51	14	CA
82	40	DB	09	57	AF	DB	0C
5F	D5	CD	BD	07	7E	FE	1D
CA	BD	40	4E	06	00	21	13
41	09	7E	32	E4	40	CD	BF
40	C1	DB	09	67	AF	DB	0C
6F	B7	ED	42	C2	82	40	C5
C3	A8	40	C1	C9	21	FF	02
01	00	00	ED	5B	E4	40	E5
D3	FF	CD	DB	40	DB	FE	CD
DB	40	E1	B7	ED	52	30	EF
C9	2A	E4	40	37	ED	42	30
FB	C9	FA	00	2A	10	40	01
08	00	09	23	7E	FE	00	28
14	FE	FF	C8	32	E4	40	23
4E	E5	C5	CD	BF	40	C1	E1
0D	20	F6	18	E6	23	4E	3E
FF	3D	20	FD	0D	28	DC	18
F6							

A számítógépprogram nem kívánságaid,
hanem utasításaid szerint működik!

```
   1 REM LN █ ???=ASN █RND<=%?█<
=£?STR$ LN █▞?▘ 5<INKEY$%RNDAT <
=%?█<=£?█ GOSUB PITAB █RNDVAL ?█
RNDAT TAN 5 COPY ▟     GOSUB ? SL
OW RND FAST PEEK COPY LN <=RND<=
 RETURN LN <=RND LPRINT █ GOSUB
?K LOAD TAN E SLOW RNDR GOSUB PI
K CLS TAN IF  E(RND| ▒ %7 COPY CO
S M SLOW RND7? FAST VAL LN █RNDA
T LPRINT $4 PLOT / NEW 77Y COPY
X4 CLEAR $C>=/ PLOT ZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZZZ
  10 PRINT "A BILLENTYU KODJA?"
  20 INPUT A
  30 PRINT A
  40 PRINT "A HANGMAGASSAG?"
  50 INPUT B
  60 IF B=0 THEN GOTO 500
  70 POKE 16659+A,B
  80 CLS
  90 GOTO 10
 100 STOP
 150 FAST
 160 RAND USR 16514
 170 GOTO 500
 300 CLEAR
 310 LET X$=""
 330 PRINT "A HANGMAGASSAG?"
 340 INPUT B
 350 PRINT B
 360 LET X$=X$+CHR$ B
 370 IF B=255 THEN GOTO 500
 380 PRINT "AZ IDO?"
 390 INPUT B
 400 LET X$=X$+CHR$ B
 410 CLS
 420 GOTO 330
 450 FAST
 460 RAND USR 16614
 500 SLOW
 505 PRINT
 510 PRINT "     LEHETOSEGEK:"
 520 PRINT
 530 PRINT "1. HANGOLAS"
 540 PRINT "2. SZINTETIZATOR"
 550 PRINT "3. DALLAM BEIRASA"
 560 PRINT "4. DALLAM LEJATSZASA

 565 PRINT "5. MEGORZES"
 570 LET Z$=INKEY$
 580 IF Z$="" THEN GOTO 570
 585 CLS
 590 GOTO (VAL Z$-1)*150
 600 SAVE "SZINTETIZATO█"
 610 GOTO 500
```

A számitógép megbizhatatlan,

de az ember még inkább.

(Gilb megbizhatatlansági törvényeiből)

II. TÁBLÁZAT

A BILLENTYŰ KÓDJA	A HANG-MAGASSÁG	A BILLENTYŰ KÓDJA	A HANG-MAGASSÁG
63	255	27	126
56	241	25	112
61	227	49	112
41	214	228	106
40	202	15	99
59	191	24	93
44	180	226	88
39	170	23	83
45	160	216	78
51	151	19	73
47	142	22	69
50	134	18	65
14	126	21	61
48	126	26	61
225	119		

SHIFT segítségével. Az egyes lenyomásával kapjuk vissza a választási lehetőségeket.
A klaviatúra felső két sora tetszés szerint hangolható, és az alsó két sor megváltoztatható. A billentyűkódot a PRINT CODE utasítással kaphatjuk meg. (Pl. PRINT CODE "A"=38).
A hangmagasság 1-től 255-ig változhat, a legmélyebb hangot jelöli a 255.
A választási lehetőségek között a 3-as és 4-es gomb egy dallam beírására és lejátszására szolgál. Kompozíciónkat a programmal együtt meg is őrizhetjük. A dallam beírásakor a hangmagasságot és az időtartamot egy 1 és 254 közötti egész számmal adhatjuk meg. Ha 0-t írunk hangmagasságként, akkor a megadott időegység hosszúságú szünetet kapunk. Ha 255-öt írunk (hangmagasságként), akkor ez a dallam beírásának végét jelzi.

Uzonyi Tamás

BESZÉLŐ ZX

Hiszik vagy nem hiszik, Csillag Péter megtanította a ZX 81-et beszélni! Hogy hogyan, ez kiderül, ha elolvassák az alábbiakat.

A BIT-LET 3. számában olvashattuk, hogyan lehet a ZX 81-et zenére bírni. ZX 81-ünk azonban ennél sokkal többre is képes. Tetszőleges hanghatásokat (beszéd, zene stb.) képes memóriájában elraktározni, és azt bármikor megismételni. A program gépi kódban íródott, amely néhány BASIC sorral van kiegészítve. A gépi kódú részt a 0 REM sor tartalmazza. Ennek betöltése a program első részének futtatásával történik. Figyelem! A programot 1 kB-os gépbe írjuk be, különben az nem fog a tárban elférni. A betöltőprogram első sorába a REM után 80 db Z betűt írjunk. Ennek helyére kerül majd a gépi kódú program. A hexadecimális számokat egyesével írjuk be a táblázat alapján. Ezután töröljük ki a 2–6. sorokat, írjuk be a POKE 16510,0 és FAST parancsokat, majd a programot vegyük fel kazettára. A program futtatásához csatlakoztassuk

```
   1 REM ZZZZZZZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZZZZZZZZ
   2 FOR I=16514 TO 16593
   3 INPUT A$
   4 POKE I,16*CODE A$+CODE A$(2
)-476
   5 NEXT I
   6 STOP
  10 RAND USR 16532
  20 PAUSE 4E4
  30 RAND USR 16563
```

ZX 81

gépünkhöz a 16 kB-os memóriát, írjuk be sorrendben a következő parancsokat:

POKE 16388, 0
POKE 16389, 66
NEW

majd olvassuk be a kazettára felvett programot. Az eltárolandó jelet a számítógép EAR feliratú aljzatán keresztül vezessük be. Ez célszerűen egy kazettára felvett szöveg legyen. A program RUN-nal indul, ekkor kb. 15 másodpercig tart a beolvasás. Közben a képernyőn hasonló csíkokat kell látnunk, mint amikor programot töltünk be. Ezután a gépi MIC kimenetére csatlakoztassunk egy erősítőt, s nyomjunk meg egy tetszőleges gombot. Az erősítőben meg kell szólalni az eltárolt hangnak. A hangminőség természetesen nem valami csodálatos. Röviden talán így lehetne jellemezni: „érthető".

Végül néhány információ azoknak, akik programomat tovább szeretnék fejleszteni. A 16532–16562 címeken található a beolvasó, a 16563–16593 címeken a visszajátszó programrész. A 16533–16534, illetve 16564–16565 címeken található 4200H címtől kezdődik a betöltött hang tárolása. A 16559 és 16590 címeken található 80H jelzi a tároló végét. Mind a két programrészlet tartalmaz egy-egy időhúzó ciklust. Ezeknek hosszát a 16550 és 16581 címeken található byte-ok állítják be. Ezeknek átírásával gyorsíthatjuk vagy lassíthatjuk a gépben tárolt hang lejátszását, valamint módosíthatjuk a tárolt hangminta hosszát.

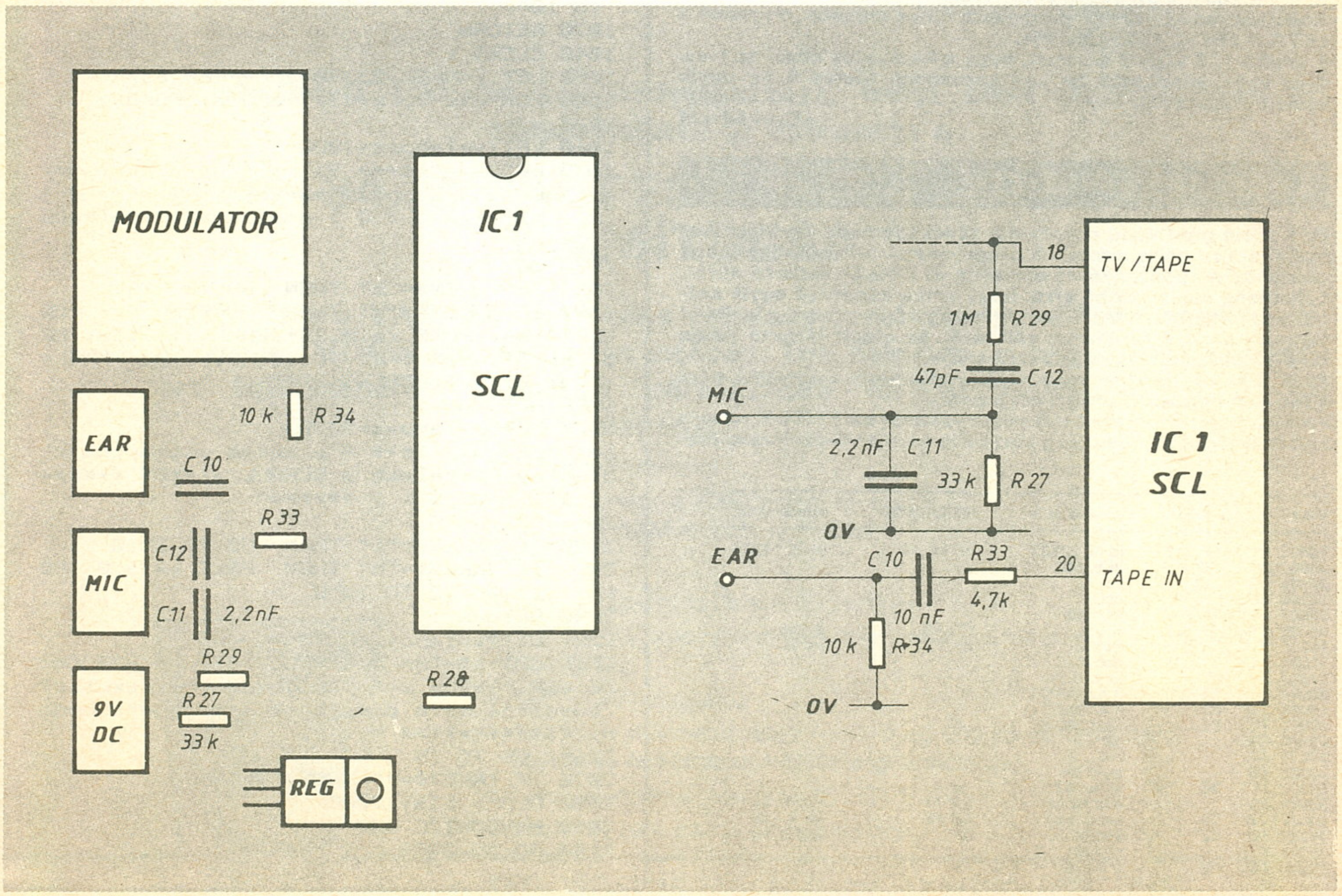
```
80 80 80 A8 9B 80 A8 B8
A1 B1 B1 A6 AC 80 80 80
76 76 21 00 42 1E 00 37
CB 13 38 0D DB FE 17 38
02 D3 FF 06 19 10 FE 18
EF 73 23 7C FE 80 20 E5
C9 21 00 42 5E 37 CB 13
28 0F 30 04 DB FE 18 02
D3 FF 06 19 10 FE A7 18
ED 23 7C FE 80 20 E5 C9
```

BIZTONSÁGI KAPCSOLÁS

Jelentkezett nálunk egy gm, kölcsönös előnyökkel járó üzletet ajánlva. Nekik jól jön egy kis publicitás, olvasóinknak pedig jól jön az a kis hasznos ötlet, amit közölni kívánnak. Íme tehát a MICROTEAM GM (Bp. XVI., Gelléri A. E. u. 40/a) ajánlata a ZX 81 kazettás tárolásának megbízhatóságát növelő kapcsolási rajz.

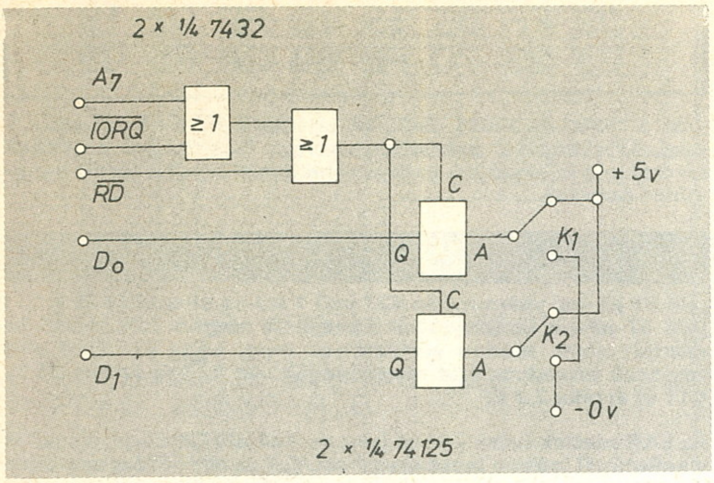
A ZX 81-es gépek kazettás tárolásának megbízhatósága sajnos javításra szorul. Az itt leírt egyszerű módosítás után, melyet bárki saját maga is elvégezhet, számottevően jobb eredményre számíthatunk. A módosítás mind a kimeneti, mind a bemeneti áramköröket érinti. A MIC jelű csatlakozóra az SCL 16-os lábán megjelenő jel van kivezetve egy osztón keresztül, mely osztót az R29 jelű 1 MΩ-os, a C12 jelű 47 pF-os, valamint az R27 jelű 1 kΩ-os ellenállás és a C11 jelű 47 nF-os kondenzátor alkotja. A magnó felé menő jel szintjét olyan módon változtatjuk meg, hogy az osztó alsó tagjának impedanciáját megnöveljük. Az R27 új értéke 33 K, C11 új értéke 2,2 nF.

Az EAR csatlakozóra érkező jelet az R 34 jelű 220 Ω-os ellenállás a kelleténél jobban lecsökkentheti. Ezt az ellenállást cseréljük ki 10 kΩ-osra, így a számítógép bemeneti érzékenysége javulni fog. A beültetési rajz részlete a nálunk legelterjedtebb ISSUE 1 sorozatra érvényes. Az itt leírt módosítással kapcsolatban, vagy más ZX 81-et, illetve ZX Spectrumot érintő kérdésben bárkinek készséggel adunk tanácsot, hardver javításokat vagy ezt a módosítást gyorsan elvégezzük.



INPUT PORT ZX-HEZ

A BIT-LET-ben a Hardverötletek között megjelent egy leírás, hogyan lehet ZX-szel két jelfogót vezérelni. Az alábbiakban szeretném megmutatni, hogyan lehet a ZX-be bevinni automatikusan adatokat.
A számítógéphez kapcsolhatunk külső érzékelőket, melyek adatai a gépbe kerülnek, és a program ezeket fölhasználja.
A bemenő és kimenő kapu egyidejű használatával tetszés szerinti folyamatokat vezérelhetünk.
Az adat-BUS-ra vezérelt ismétlő kapuval kell csatlakoznunk, állandó kapcsolat esetén sem zavarja a BUS működését. (74 125)



A kapcsolási rajzon egy kétbites input port látható, de mivel az IC-ben 4 kapu van, egyszerűen kiterjeszthető négy bitre.
Az információk beolvasása ZX 81 esetén a következő gépi programmal történhet: 219, 127, 6, 0, 79, 201, melyet szokásos módon az első REM sorba írunk be.
A kapcsolók állásának leolvasása ZX 81-nél:
LET T = 255 – USR 16514 utasítással történhet.
Spectrum esetén erre a programra nincs szükség, elegendő a LET T = 255 – IN 127 utasítás.
T értéke 1, 2 vagy 3 aszerint, hogy a kapcsolók közül az 1-es, 2-es vagy mindkettő be van kapcsolva, illetve 0, ha egyik sincs bekapcsolva.

Futó László

ASSEMBLER RUTIN

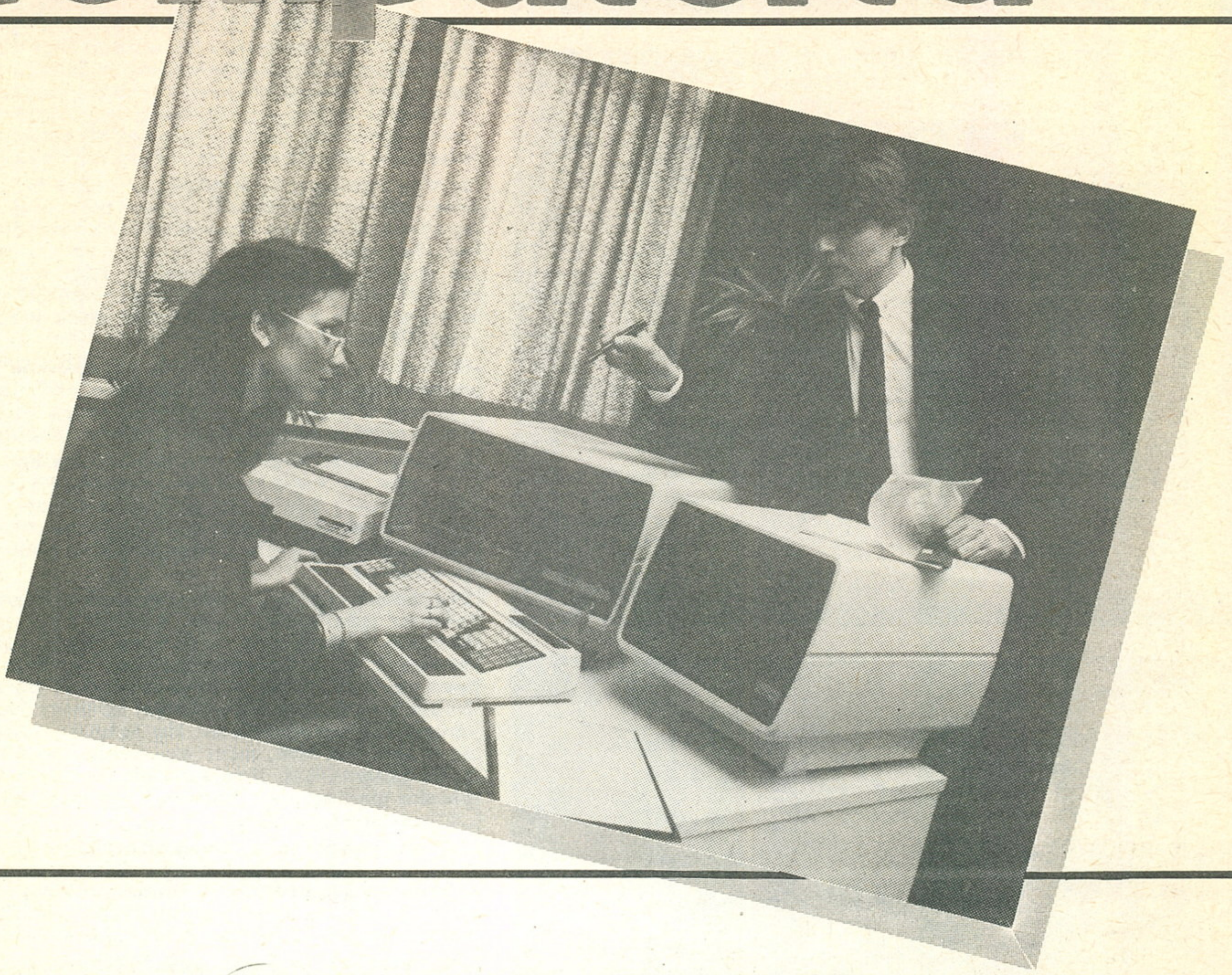
Ez a rövid gépi kódú program a tévéképernyőn levő ábrákat tudja jobbra-balra eltolni. A program specialitása még, hogy pixelenként (pontonként) mozdítja el a tévén látható ábrát. Lehetőség van továbbá arra, hogy ami a képernyő egyik szélén kitolódik, az a másikon visszajöjjön. A program a RAM bármely szabad területére tölthető. Az oldalt látható listán én a 30000-es tárcímet választottam.
Ez a lista ne ijesszen meg senkit, az ennek megfelelő gépi kódú részt a BASIC program „megírja".
Az assembler program beírásához valamilyen EDITOR rutinra van szükség, különben elég körülményes a hexadecimális számokat átszámítani. Ehhez a beíráshoz szeretnék segítséget nyújtani az alábbi programmal. Ez a program olyan, mint a papírtörülköző: felhasználás után eldobható.
Miután beírtad a BASIC programot, a RUN paranccsal indítsd el, válaszolj a kérdésekre, ezután láthatod, mire képes ez a SCROLL rutin. Ezek után BREAK-kel lehet megállítani a programot. NEW-val lehet törölni a BASIC programot. A gépi kódú rész persze megmarad. Ezt a 48 kbyte-os gépen az alábbi programmal próbáltam ki:
10 CIRCLE 126, 87, 87
20 RANDOMIZE USR 65501
30 GOTO 20

Szenttornyai László

CÍM	HEXA-KÓD			mnemonik	HEXA-KÓD	mnemonik
30000	21	FF	57	ld hl, 22527	21 00 40	ld hl, 16384
30003	06	CO		ld b, 192		
30005	37			scf		
30006	3F			ccf		
30007	16	20		ld d, 32		
30009	CB	16		rl (hl)	CB 1 E	rr (hl)
30011	2B			dec hl	23	inc hl
30012	15			dec d		
30013	7A			ld a, d		
30014	20	F9		jr nz, 30009		
30016	38	03		jr c, 30021		
30018	10	F1		djnz 30005		
30020	C9			ret		
30021	3F			ccf		
30022	C5			push bc		
30023	01	20	00	ld bc, 32		
30026	09			add hl, bc	ED 42	sbc hl, bc
30027	CB	C6		set 0, (hl)	CB FE	set 7, (hl)
30029	ED	42		sbc hl, bc	09	add hl, bc
30031	C1			pop bc		
30032	18	FO		jr 30018		

```
   0>DATA 30,8,33,31,71,6,8,55,63,22,32,
203,22,43,21,122,32,249,55,63,197,1,224,
0,237,66,193,16,234,29,123,32,225,201
  10 DATA 33,255,87,6,192,55,63,22,32
  20 DATA 203,22,43,21,122,32,249,56
  30 DATA 3,16,241,201,63,197,1,32,0
  40 DATA 9,203,198,237,66,193,24,240
  50 POKE 23756,0: POKE 23617,0
  55 DATA 237,66,203,254,9
  60 LET valahova=1000
  70 LET a$="****** Ha 16 Kbyte-os Spect
rumod van,akkor nyomd meg a { Z } gombot
,ha 48 Kbyte-os akkor a { X } gombot! **
****"
  75 RESTORE : FOR n=23263 TO 23296: REA
D a: POKE n,a: NEXT n
  80 FOR n=1 TO LEN a$
  90 PRINT AT 0,31;a$(n)
  95 GO SUB valahova
 120 LET a=USR 23263: NEXT n: GO TO 75
1000 IF INKEY$="z" THEN  LET x=32730: GO
 TO 1040
1010 IF INKEY$="x" THEN  LET x=65500: GO
 TO 1040
1020 RETURN
1040 CLEAR x
1050 LET x=PEEK 23730+256*PEEK 23731+1
1055 FOR m=x TO x+33: READ a: POKE m,a:
NEXT m
1060 LET valahova=1080
1070 LET a$="***** Balra vagy jobbra moz
duljon el a display tartalma? ****** A {
 B } illetve a { J } gombokkal felelhets
z. ****"
1075 GO TO 75
1080 IF INKEY$="b" THEN  GO TO 2000
1090 IF INKEY$="j" THEN  POKE x+1,0: POK
E x+2,64: POKE x+10,30: POKE x+11,35: RE
STORE 55: FOR f=26 TO 30: READ a: POKE x
+f,a: NEXT f: GO TO 2000
1095 RETURN
2000 LET valahova=2020
2010 LET a$="**** A program megtartsa a
display tartalmat a SCROLL alatt? ******
 { I } vagy { N } ******"
2015 GO TO 75
2020 IF INKEY$="i" THEN  GO TO 2050
2030 IF INKEY$="n" THEN  POKE x+16,0: PO
KE x+17,0: GO TO 2050
2040 RETURN
2050 LET valahova=2070
2060 LET a$="** A SCROLL rutin a RANDOMI
ZE USR "+STR$ x+" parancsra egy pixellel
 mozditja el a display-t a megadott modo
n. ************ "
2065  GO TO 75
2070 IF INKEY$="" THEN  RETURN
2080 CLS : LIST 2000
2090 RANDOMIZE USR x
2100 GO TO 2090
```

Computerta



A népszerű TAP–34 továbbfejlesztett, helyi adatfeldolgozásra jól alkalmazható új változata a

TAP—34/M PROFESSZIONÁLIS SZEMÉLYI SZÁMÍTÓGÉP

biztosítja a TAP–34 operációs rendszer, valamint a CP/M-mel kompatibilis VDOS rendszer futtatását. Így minden, a régi berendezésre írt felhasználói program változtatás nélkül használható az új berendezésen.

Műszaki paraméterek:

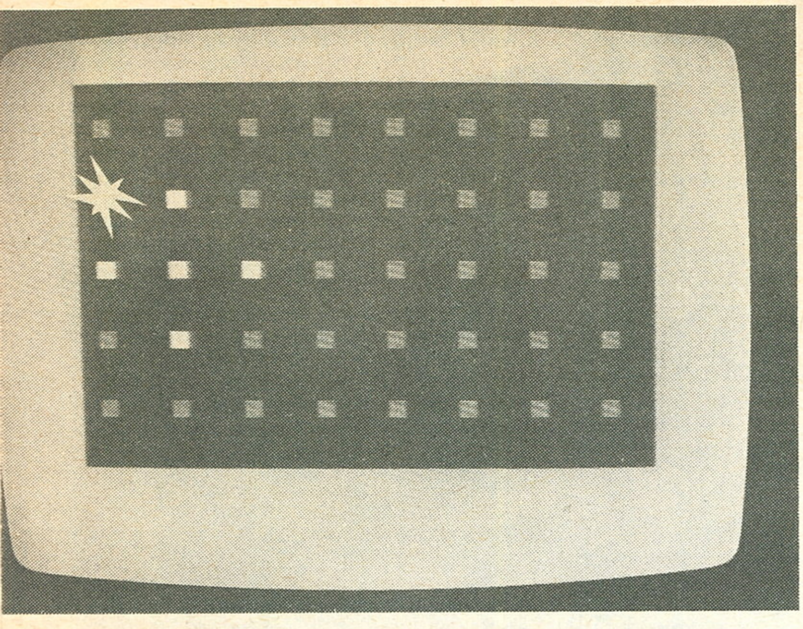
Vezérlő egység: • 8080 mikroprocesszor
• 64 kbyte RAM • 4 kbyte PROM
Perifériák:

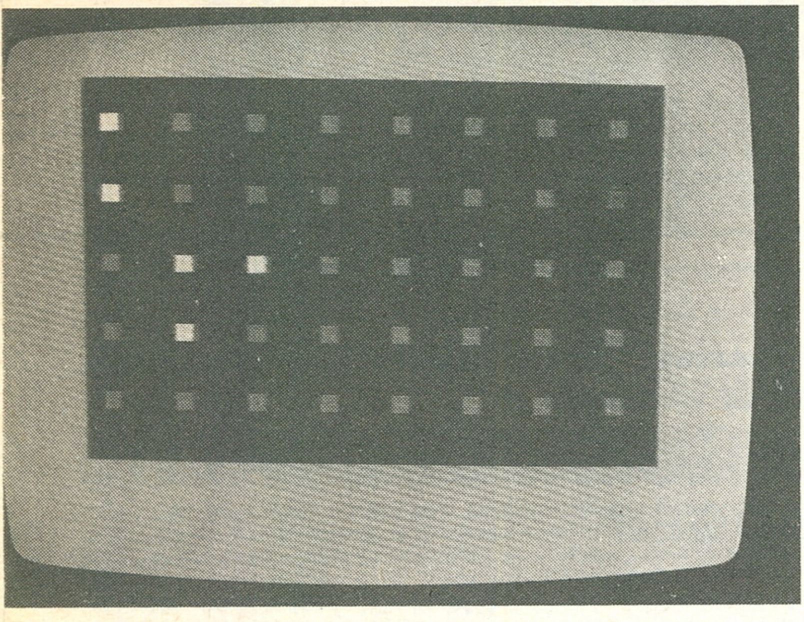
- 12"-os képernyő, 25 sor, 80 karakter
- hall generátoros billentyűzet, alfanumerikus, numerikus és szerkesztőbillentyűk,
- TMT–120 nyomtató,
- hajlékony mágneslemezek (max. 4 db 600 kbyte kapacitású, egyszeres és kétszeres írássűrűségre alkalmas),
- soros adatátviteli csatorna (CCITT V. 24)
- általános célú párhuzamos interface.

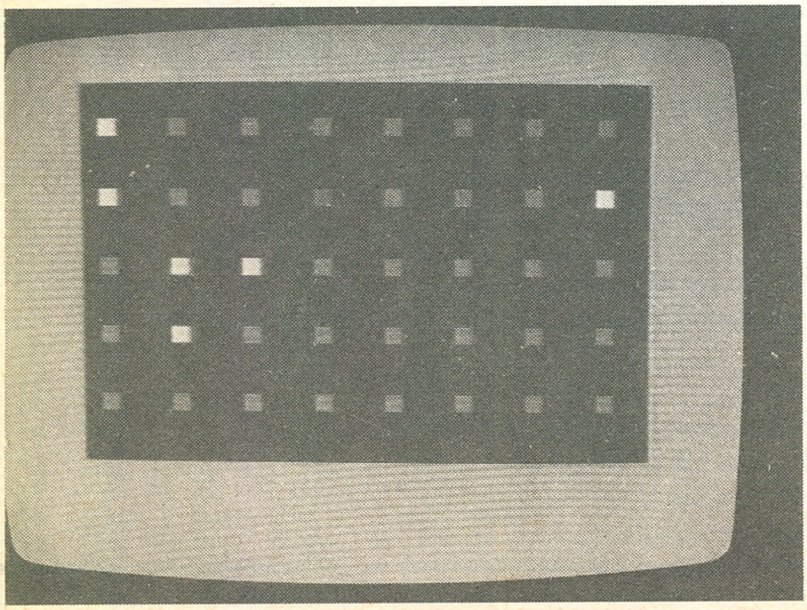
TERTA

TELEFONGYÁR 1143 Budapest, Hungária körút 126–132.

BŰVÖS LÁMPÁCSKÁK







Igazán nehéz dologra vállalkoztunk, amikor elhatároztuk, hogy egy rövid kis játékprogramot közlünk a ZX Spectrumhoz. A dolog nehézségét az tudja fölmérni igazán, aki ismeri a Spectrumhoz kapható, s ily módon legalábbis a legmegszállottabb gyűjtők körében nálunk is fellelhető játékprogramokat. Egyik cselesebb, mint a másik, ráadásul a gép grafikus lehetőségeit is a legmesszebbmenőkig kihasználják ezek a „profi" játékok. Sajnos, bele kellett törődnünk, hogy mi legfeljebb jó amatőr munkát tudunk fölkínálni.

Nem is a megvalósíthatósággal volt a baj, hanem a zseniális ötlet hiányzott. Először ugyan korcsolyás télapók cikáztak az agyunkban, meg Horatius újabb kalandjain törtük a fejünket, de végül is győzött a harcos opportunizmus, s egy kevésbé látványos, de kivitelezhető ötlet megvalósítása mellett döntöttünk. Követtük a hazai játékipar jól bevált irányzatát, s egyszerű, de nagyszerű logikai játék kidolgozásával bíztuk meg Király Zolit.

Játékunk a jól hangzó „bűvös lámpácskák" nevet kapta. Látszólag nagyon egyszerű feladat elé állítja a játékost, ám a gyakorlati próba során kiderül, hogy nem is olyan könnyű dolog néhány lámpácskánkba életet lehelni.

A játékosnak először arra a kérdésre kell válaszolnia, hogy „a" vagy „b" variánssal akar-e játszani. („a" és „b" jelentéséről később.) Majd azt kell elhatároznunk, hogy hányszor hány lámpácskát akarunk kirajzoltatni képernyőnkre. (A vízszintes oldalon 2–15, a függőlegesen pedig 2–10 között lehet ez a szám.)

Döntésünk alapján a képernyőn megjelenik az általunk választott számú lámpácska. Közülük egy sem világít az induló állásban. Feladatunk éppen az, hogy valamennyibe életet leheljünk. Ez a következőképpen történik: A billentyűzeten levő négy nyíl (5, 6, 7, 8) segítségével bármelyik lámpácskát kiválaszthatjuk. A kiválasztást az mutatja, hogy a megfelelő lámpácska villogni kezd. Ha helyben vagyunk, semmi mást nem kell tennünk, mint megnyomni a SPACE vagy az 1-es szám billentyűjét. Erre azonban nemcsak a kiválasztott lámpácska gyullad meg, hanem... – s most következik az „a" és „b" variáns jelentősége.

„a" játék esetében a kiválasztott lámpa és függőleges, valamint vízszintes irányú szomszédai is meggyulladnak. Tehát a belül levő lámpáknál ez önmagán kívül még négy szomszédot, az alsó és felső sorban levők esetében pedig hármat-hármat jelent. (A sarkon levőknél csak kettőt.)

„b" játékban azonban egy fokkal bonyolultabb a helyzet. Képzeljük el, hogy a téglalap alaprajzú játékmező szemközti széleit egymáshoz hajlítjuk. Olyan az egész, mint egy úszógumi vagy autóbelső. Ily módon tehát a szélen levő lámpácskáknak is mindig négy szomszédja lesz! (Erre szokták mondani, hogy „hoppá".)

Ez így önmagában még nem tenné túl nehézzé a játékot, csakhogy amelyik lámpácskát meggyújtottuk, el is olthatjuk. Ha ugyanis olyan lámpa van a kiválasztott és szomszédai közt, amelyik már ég, akkor ez bizony elalszik. A lámpácskák tehát mindig az ellenkezőjükre változnak.

Jól példázza a játék lényegét az itt látható három képernyőfotó. Az 1. számú alaphelyzetből indulunk. Amint látjuk, a bal oldali lámpácskánk különbözik mindegyiktől. Ez jelzi, hogy „őt" választottuk. A 2. képen láthatjuk, hogy mi történik a SPACE gomb megnyomása után az „a" játékban, s a 3. képen pedig, hogy mi a „b" játékban. A kiválasztott lámpácskák szomszédai közül az, amelyik világított, elalszik, amelyik pedig nem volt meggyújtva, fölgyullad.

A program beírásához két rövid használati utasítás. Az egyik a fekete-fehér tévével rendelkezőknek szól. Javasoljuk, hogy a jobb láthatóság érdekében a 40-es sorba BO = 7, PA = 0, BR = 0, S1 = 4, S2 = 7, S3 = 7 adatokat írják be. A másik apróság az ékezetes betűkkel kapcsolatos. A program ismeri az „Á, Ö, Ü" betűket, sajnos a nyomtatónk nem. Azt javasoljuk, hogy az 5-6-7-es sorok leírása után futtassuk le a programot, s akkor már használhatjuk ezeket a betűket. Például ahol a szövegben „Ö" betű kell, ott átváltunk grafikus üzemmódba, s úgy nyomjuk le az „O" betűt. Hasonlóan kaphatunk az „A"-ból „Á"-t, vagy az „U"-ból „Ü"-t.

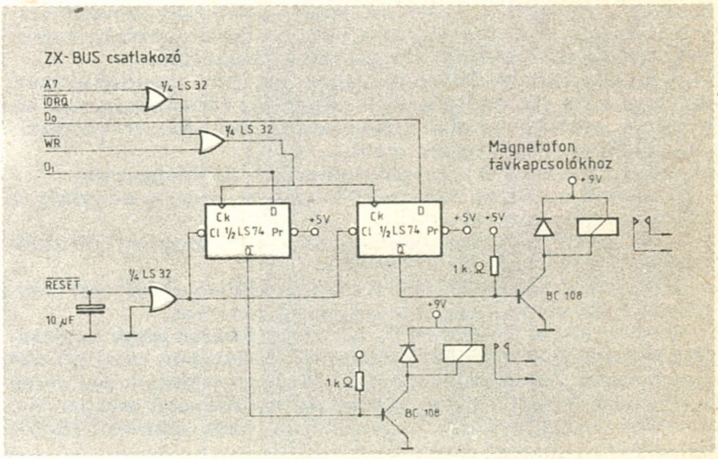
És még egy. Tudjuk, hogy nem szép ilyet feltételezni olvasóinkról, de ha mégis föl akarják adni a játékot, csak nyomják meg az „f" billentyűt. Mire a gép megkérdezi majd, hogy akarják-e elölről kezdeni. Ha igent válaszolnak, akkor ugyanazt a mezőt kapják vissza kiinduló helyzetben, ha nem, még mindig két lehetőségük van. Vagy új adatokat adva folytatják a játékot, vagy elmennek inkább aludni.

Szép álmokat!

```
 10 REM ( KESZITETTE KIRALY ZOLTAN 1983-BAN.
 )
 20 DATA 40,40,16,40,0,16,68,124,56,68,68,68
,68,68,124,68,68,68,124,124,68,0,0,0
 30 FOR I=0 TO 7: READ A,B,C: POKE USR "u"+I
,A: POKE USR "o"+I,B: POKE USR "a"+I,C: NEXT
I
 40 LET BO=6: LET PA=0: LET BR=1: LET S1=3:
LET S2=7: LET S3=6
 50 DIM X(5)
 60 DIM Y(5)
 70 BORDER BO: PAPER PA: BRIGHT BR: INVERSE
1: CLS
 80 DEF FN W(X,Y)=ATTR (C+N*X,B+M*Y)-8*INT (
ATTR (C+N*X,B+M*Y)/8)
 90 INK S3: PRINT AT 10,7;"BUVOS LAMPACSKAK"
 100 INPUT "AZ 'A', VAGY A 'B' VALTOZATTAL  A
KARSZ JATSZANI?";Z$
 110 IF Z$<>"A" AND Z$<>"a"  AND Z$<>"B" AND
Z$<>"b" THEN  GO TO 100
 120 LET P=0: INPUT "A HOSSZABB OLDAL?";Q$: G
O SUB 600: IF Q<2 OR Q>15 OR KK=1 THEN  GO TO
 120
 130 LET P=Q
 140 INPUT "A ROVIDEBB OLDAL?";Q$: GO SUB 600
: IF Q<2 OR Q>10 OR KK=1 THEN  GO TO 140
 150 LET M=INT (32/P): LET B=15-(P+1)/2*M
 160 LET N=INT (22/Q): LET C=10-(Q+1)/2*N
 170 CLS : INK S1: FOR I=1 TO P: FOR J=1 TO Q
 180 PRINT AT C+N*J,B+M*I;" ": NEXT J: NEXT I
: LET CX=1: LET CY=1
 190 LET O=FN W(CX,CY): INK O: PRINT  FLASH 1
;AT CX*N+C,CY*M+B;" "
 200 LET A$=INKEY$: IF A$="" THEN  GO TO 200
 210 IF A$="F" OR A$="f" THEN  GO TO 300
 220 IF A$=" " OR A$="1" THEN  GO TO 400
 230 IF A$<"5" OR A$>"8" THEN  GO TO 200
 240 PRINT AT CX*N+C;CY*M+B;" "
 250 LET A=VAL A$
 260 LET CX=CX+(A=6)*(CX<Q)-(A=7)*(CX>1): LET
 CY=CY+(A=8)*(CY<P)-(A=5)*(CY>1): GO TO 190
 300 INPUT "ELOLROL AKAROD KEZDENI?";A$: IF A
$(1)="I" OR A$(1)="i" THEN  GO TO 170
 310 INPUT "AKARSZ TOVABB JATSZANI?";A$: IF A
$(1)="I" OR A$(1)="i" THEN  GO TO 100
 320 BORDER 7: PAPER 7: BRIGHT 0: INVERSE 0:
INK 0: CLS : STOP
 400 FOR I=1 TO 5: LET X(I)=CX+(I=5)-(I=4): L
ET Y(I)=CY+(I=2)-(I=3): NEXT I
 410 IF Z$="A" OR Z$="a" THEN  GO TO 460
 420 LET X(4)=X(4)+(X(4)<1)*Q
 430 LET X(5)=X(5)-(X(5)>Q)*Q
 440 LET Y(2)=Y(2)-(Y(2)>P)*P
 450 LET Y(3)=Y(3)+(Y(3)<1)*P
 460 FOR I=1 TO 5
 470 IF X(I)<1 OR X(I)>Q OR Y(I)<1 OR Y(I)>P
THEN  GO TO 490
 480 INK S1+S2-FN W(X(I),Y(I)): PRINT AT X(I)
*N+C,Y(I)*M+B;" "
 490 NEXT I
 500 FOR I=P TO 1 STEP -1: FOR J=Q TO 1 STEP
-1
 510 IF FN W(J,I)<>S2 THEN  GO TO 190
 520 NEXT J: NEXT I
 530 PRINT  INK S3; FLASH 1;AT 20,5;"GRATULAL
OK, SIKERULT!!!"
 540 GO TO 310
 600 LET KK=0: IF LEN Q$=0 OR LEN Q$>2 THEN
GO TO 630
 610 LET Q$=("0"+Q$)(LEN Q$ TO )
 620 IF Q$>"0" AND Q$<"99" AND Q$(2)>="0" AND
 Q$(2)<="9" THEN  LET Q=VAL Q$: RETURN
 630 LET KK=1: RETURN
 999 SAVE "lampa" LINE 10
```

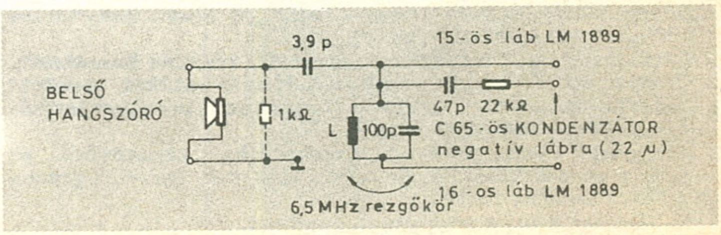
TÁVKAPCSOLÓ

ZX-es körökben a központi téma legtöbbször a gép bővíthetősége. A most bemutatott kapcsolási rajz egy kétbites output port, mellyel két magnetofont (vagy bármi más kisfeszültségű fogyasztót) lehet a programunkból vezérelni. A kapcsolóelemek miniatűr 9 V-os relék. A kazettás magnetofonok távirányításához még a legolcsóbb típusok is megfelelnek.
Az output port a CPU sínén, a 127-es port-címen található, melyet a ZX működése közben ngyébként nem használ.
ZX 81 esetén a mozgatáshoz egy ötbyte-os gépi kódú programrész szükséges, mely a szokott módon (lásd BIT-LET 1. szám) az első REM sorba kerül. Két utasítást kell ezután használni, ha meg akarjuk változtatni vezérlőnk állapotát:
POKE 16515,x
RAND USR 16514
Az ötbyte-os program decimális értékei:
62, 0, 211, 127, 201
SPECTRUM-nál egyszerűbb az életünk:
OUT 127,x
Ha x értéke 1, 2, 3, akkor rendre az első, második vagy mindkét magnót bekapcsolja, vagy x lehet 0, s ekkor mindegyiket kikapcsolja számítógépünk.
Rauscher Attila



SPECTRUM-HANG A TV-HANGSZÓRÓN KERESZTÜL

A ZX Spectrum hangját sokan csak a beépített hangszórón szokták hallgatni, bár lehetőség van arra is, hogy a MIC csatlakozó jelét erősítőre vezessük, és így külső hangszóróra kapcsoljuk.
Az alábbiakban közölt áramkör segítségével a hangot az antennajelre keverhetjük, és így már nemcsak a képet, de a hangot is a televízión keresztül fogjuk kapni, ugyanúgy, mint a Commodore számítógépnél.



A Spectrumba épített LM 1889 típusú „videó-IC" alkalmas a hangjel fogadására is, de ezt a gyári gépeknél nem használják ki; az IC 15-ös lábára +12 V-ot kapcsolnak. Válasszuk le a 15-ös lábat a +12 V-os feszültségről, és kössük be a rajznak megfelelően. A kapcsolásban található rezgőkört 6,5 MHz frekvenciára kell hangolni a tekercs vasmagjának finombeállításával. A hangolást úgy végezzük el, hogy a televízió hangszóróján zajminimumot kapjunk.
A számítógép belső hangszóróját akár ki is kapcsolhatjuk, de ebben az esetben a helyére egy 1 kohm-os ellenállást kell beforrasztani.
Bankó Miklós MICROTEAM GM

FÉNYKÉPEZŐGÉP

A személyi számítógépek egyik legvonzóbb sajátsága az egyszerű és „gyors" grafika. Ennek élvezetéhez nyújtunk egy kis segítséget, elsősorban a HT adottságokat figyelembe véve.

Az 1. sz. program 45.–46. sorában vannak az ábra kódjai. A rajz minden egyes „pontjáról" közöljük, merre van az előzőhöz képest.

A B C
H □ D
G F E

Tehát DFA jelentése: egyet jobbra, egyet lefelé, egyet balra fel (azaz ugyanoda jutottunk vissza!).

A számok segítségével rövidíthetünk, 6H jelentése: H H H H H H (vö. EDIT parancs). U betű felemeli a „tollat", majd ismét utána következő két betűt nyom nélkül teljesíti, majd ismét rajzol. Tehát a 45. sor első karakterei hatására (U6D7F) ábránk a (6,7) pontban kezdődik.

Ha egy kép nem fér el kényelmesen egyetlen DATA sorban, csillag segítségével újabb sorban folytatható.

Az említett karakterek értelmezése a program lényege:
203 – *-ot keres, hatására újabb adatot olvas.
205 – a számokat ismeri fel, természetesen többjegyűeket is.
600 – képernyőn kívülre írás ellen véd.
620 – ún. dupla pontokat rajzol.
100–110 sorokban az A–H betűknek megfelelő elmozdulásvektorok koordinátáit olvassuk be. Ez sokkal gyorsabb rajzolást eredményez, mint az IF-es esetszétválasztás.

Ha végrehajtjuk a 2. programban látható módosításokat, akkor az ábra 90° egész számú (véletlen) többszöröseivel elforgatva jelenik meg. Érdemes megfigyelni a képernyő torzítását! Idővel ezt ki fogjuk védeni.

Aki az eddigieket érti, elgondolkodhat az alábbiakon:

1. Próbáljuk az ábrát nagyítani – feltéve, hogy a nagyítás is „ráfér" a képernyőre!
2. Lehet-e 45° (páratlan) többszöröseivel is forgatni? Próbáljuk ki!
3. A 46. sor elején FGFGFGFGFG-t próbáljuk 5(FG)-vel rövidíteni! Nem könnyű, és nem is mindig érdemes.
4. További képek készíthetők, ha négyzethálós papír segítségével kicseréljük a 45–46. sorokat. (A 3. listában közöljük két ábra kódjait. Érdemes kipróbálni!) Közlünk azonban egy olyan programot is, ami ezt lényegében automatikusan elvégzi. Aki azonban akarja, próbálja a programot maga megírni. (Érdekes feladat.)

Ha beírjuk a 4. programot, a következőképpen dolgozhatunk: Rajzunkat négyzethálós papír helyett egy átlátszó fóliára készítsük, amit a képernyőre kell feszíteni (cellux). Ezután egy „pontot" kell végigvezérelni a rajz vonalai alatt. A pont nyomát a gép jelzi, és megjegyzi, majd ezt a kívánt stringalakban kódolja.

Konvenciók: – a Vezérlés
U I O
J □ K billentyűkkel
N M ,
történik. Jobban kézre esnek, mint az A–H betűk (210–217. sorok).

4U jelentése ismét UUUU (205. sor).

R betű felemeli a „tollat", illetve visszateszi: a következő pont már eszerint látszik vagy sem (RESET). Kiinduló állás – felemelt toll. R benyomása – rajzol, R újbóli benyomása – toll ismét fel stb. (201-es sor).

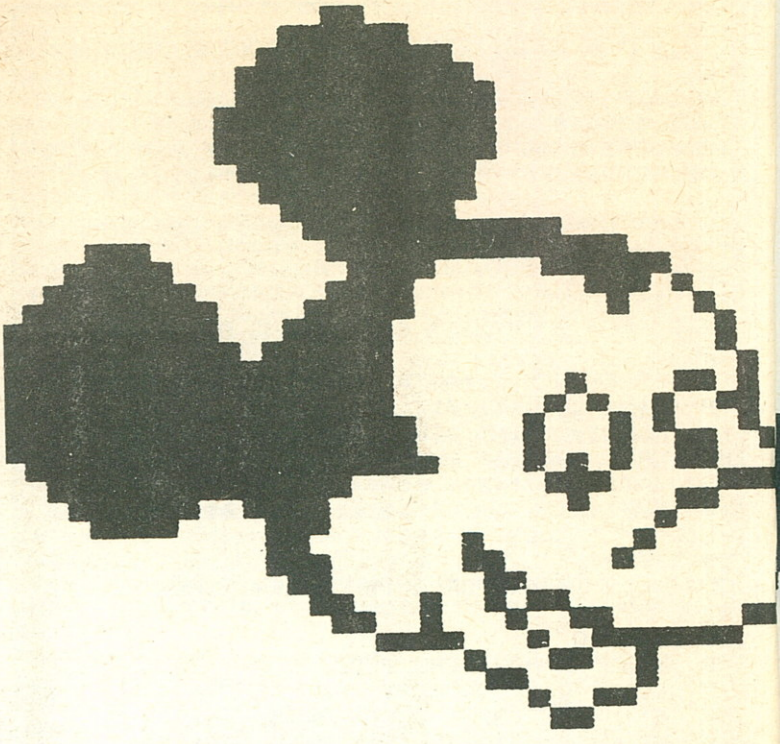
V hatására a pont mozgatása véget ér, kódolás következik, majd a képernyőn megjelennek az A–H és U betűket, valamint a számokat tartalmazó stringek. (Ezeket az 1. program természetesen könnyedén értelmezi.)

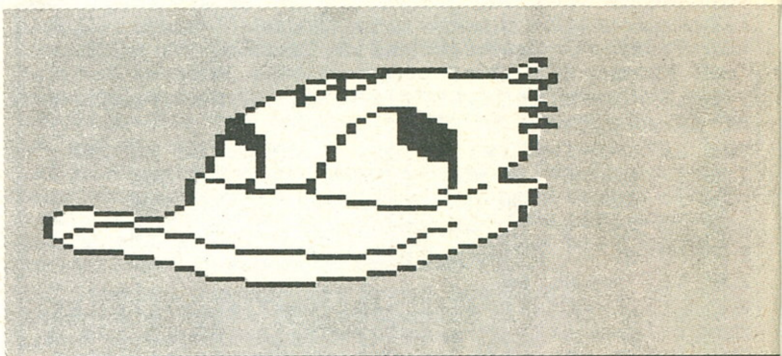
T betűvel az utolsó pont(ok) törölhető(k) a képernyőről, és persze a memóriából is. Evvel csínján kell bánni, ugrásba (R váltás) soha ne töröljünk bele!

Ezek alapján tehát a program használata:

1. Tegyük a rajzos fóliát a képernyőre – segít a képmező sarkait jelző három pont (50-es sor).
2. Mozgassuk a pontot egy vonal alá (betűk, illetve számok). (Vigyázzunk, mert induláskor a pont a képernyő bal felső sarkában van. Tehát befelé hozzuk!)
3. R benyomása után a mozgatások már nyomot hagynak. Megrajzolunk egy összefüggő darabot, majd ugrás után (R) újabb ~~darab~~ következhet stb.
4. Ha elégedettek vagyunk a művünkkel, V betűt megnyomva már csak a gépnek van dolga. 10–100 mp alatt megjelennek a stringek, amelyeket a DATA-kba írva bármikor reprodukálható az ábra. Kis gyakorlattal „bármi" elkészíthető.

Török Turul



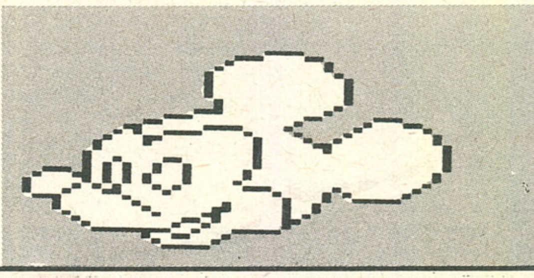


1. sz. program

```
1 CLS
3 CLEAR 1000
4 N=10
5 X9=63:Y9=47:X8=0:Y8=0:F1=(X9-X8)/2:F2=(Y9-Y8)/2
8 DATA -1,0,1,1,1,0,-1,-1
9 DATA -1,-1,-1,0,1,1,1,0
45 DATA"U6D7FD2E7D2BC2DCDED2CBC2DEDCDEF2DC2EDEF2GHF2G*"
46 DATA"FGFGFGFGFGF4HG5HGHG2HG4HAH7ABA2C2BCBA3D2BU1D11
r2DA3DA2DBDU5D7BEFG7FG6FU18D21BH3GFGH2GF2G9F"
100 FOR I=1 TO 8:READ V(I):NEXT I
110 FOR I=1 TO 8:READ F(I):NEXT I
155 READ A$
180 CLS
200 FOR J=1 TO LEN(A$)
202 A=ASC(MID$(A$,J,1)):IF A=85 THEN 500
203 IFA=42 THEN READA$:GOTO 200
205 IF ABS(A-52.5)<5 THEN F=F*10+A-48:GOTO 240
210 MA=A-64
217 FOR I=1 TO F:X=X+V(MA):Y=Y+F(MA)
218 GOSUB 600
230 NEXT I
235 F=0
240 NEXT J
295 B$=INKEY$:IF B$="" THEN 295
299 END
499 '*****UGRA'S (FELEMELT TOLL)******
500 J=J+1:A=ASC(MID$(A$,J,1))
505 IF ABS(A-52.5)<5 THEN F=F*10+A-48:GOTO 500
510 MA=A-64:X=X+F*V(MA):F=0
520 J=J+1:A=ASC(MID$(A$,J,1))
525 IF ABS(A-52.5)<5 THEN F=F*10+A-48:GOTO 520
530 MA=A-64:Y=Y+F*F(MA):F=0
580 GOSUB 600
599 GOTO 240
600 IF ABS(X-F1)>F1 OR ABS(Y-F2)>F2 THEN 690
620 SET(X*2,Y):SET(X*2+1,Y)
690 RETURN
```

2. sz. program

```
11 DATA 0,0,47,0,47,47,0,47
100 FOR I=1 TO 8:READ V1(I):NEXT I
110 FOR I=1 TO 8:READ F1(I):NEXT I
120 FOR I=0 TO 3:READ X(I),Y(I):NEXT I
170 FO=RND(4)-1
172 FORI=1 TO 8:I1=I+2*FO:IF I1>8 THENI1=I1-8
174 V(I)=V1(I1):F(I)=F1(I1):NEXT I
180 X=X(FO):Y=Y(FO)
```

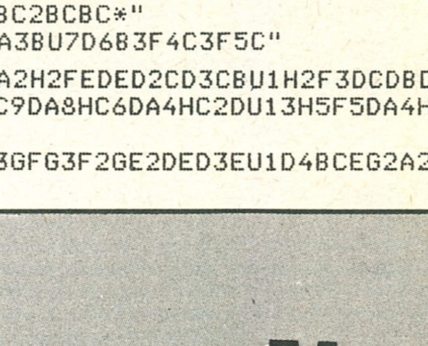


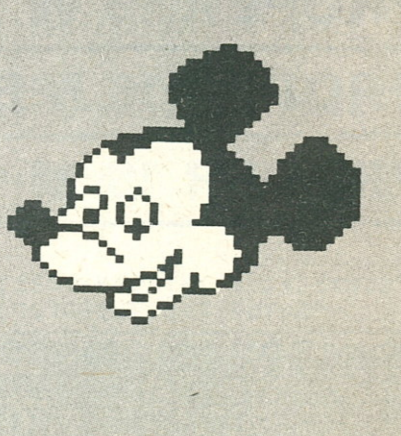
3. sz. program

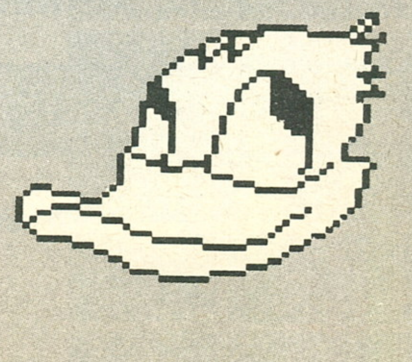
```
55 DATA "U8D36F2E4DE2DEDED2E3DE6DC4DC2DCD2CD2CD5C2BC2BCA3H2B2CBC2BAB3D3A3D2AB2CB
G3H3CBH3G5HA6HG4HGH2GH4GFGFG3FG2F2G4F3G2HA3HA2H2G3FE2CDC3DE2DE2D2E2DE6DE5DC2D4CD
CU3D7B2GH2G5HA2HA2H2A2HG2HAHAHU4D1FBC7BABAU7D12FBC2BC2BCBC*"
56 DATA "B3C3D4E10FGU0D6B5BA6FA6BA6FA5BU15H6F3BAB4FA3BU7D6B3F4C3F5C"
50 DATA "U4D26FDE3HG5DF5HE4DG2HEF3EDE5DCDCD3CFGF3GHA2H2FEDED2CD3CBU1H2F3DCDBDCHB
2DBHA2DC3HA13D2G3HA4D2CD18HC18DB18HC17DB17HB17DB10HC9DA8HC6DA4HC2DU13H5F5DA4HA4D
A3HA3DC4HB5DC6HB7DC8HB9DB9HB9DA8HB8DA7HC5DHAH*"
51 DATA "U4H2FGDF2HF2DF2HF2DFHFD3FHB4HG4DU3H1FHAHGH3GFG3F2GE2DED3EU1D4BCEG2A2B2C
2E2FU8H0FBEBCBAHGCU3D5BECBHF"
```

4. sz. program

```
3 CLEAR 500
5 X9=63:Y9=47:X8=0:Y8=0:F1=(X9-X8)/2:F2=(Y9-Y8)/2
8 DATA 0,-1,1,0,0,-1,1,-1,1
9 DATA -1,0,0,0,1,1,-1,-1,1
10 DATAA,H,G,B,"",F,C,D,E
20 DIM X(500),Y(500)
21 DIM T$(30)
40 CLS
50 SET(127,0):SET(127,47):SET(0,47)
51 X=0:Y=0
100 FOR I=1 TO 9:READ V(I):NEXT I
110 FOR I=1 TO 9:READ F(I):NEXT I
120 FOR I=1 TO 9:READ B$(I):NEXT I
180 R=1
200 A$=INKEY$:IF A$="" THEN  200
201 IF A$="R" THEN R=3-R
202 IF A$="V" THEN 400
205 A=ASC(A$):IF ABS(A-52.5)<5 THEN F=F*10+A-48:GOTO 200
210 MA=A-72:IF A=44 THEN MA=9
214 IF A=85 THEN MA=8
215 IF A=84 THEN 800
216 IF MA>9 OR MA<1 THEN 250
217 FOR I=1 TO F:X=X+V(MA):Y=Y+F(MA)
218 IF ABS(X-F1)>F1 OR ABS(Y-F2)>F2 THEN X=X-V(MA):Y=Y-F(MA):GOTO 247
220 SET(2*X,Y):SET(2*X+1,Y)
225 IF W=1 THEN RESET(X1*2,Y1):RESET(X1*2+1,Y1):X1=X:Y1=Y
230 IF R>1.2 AND(X-X(K)<>0 OR Y-Y(K)<>0)  THEN K=K+1:X(K)=X:Y(K)=Y:W=0
240 IF R<1.2 THEN X1=X:Y1=Y:W=1
245 NEXT I
247 F=0
250 GOTO 200
399 '******KO'DOLA'S******
400 PRINT@576," ";
402 B=5:X=X(1):Y=Y(1):A$="U"+MID$(STR$(X),2,3)
403 A$=A$+"D"+MID$(STR$(Y),2,3)+"F"
410 FOR I=2 TO K
412 IF ABS(X(I)-X)>1.1 OR ABS(Y(I)-Y)>1.1 THEN 500
415 A=(X(I)-X)*3+Y(I)-Y+5:IF A=5 THEN 450
420 X=X(I):Y=Y(I):IF A=B THEN F=F+1:GOTO 450
430 IF F>1 THEN A$=A$+MID$(STR$(F),2,3)
440 A$=A$+B$(B):B=A:F=1
442 IF LEN (A$)>20 THEN N=N+1:T$(N)=A$:A$=""
450 NEXT I
460 IF F>1 THEN A$=A$+MID$(STR$(F),2,3)
465 A$=A$+B$(A)
470 N=N+1:T$(N)=A$
471 FOR I=1 TO N:PRINTT$(I),:NEXT I
490 END
499 '******UGRA'S KO'DOLA'SA ******
500 IF F>1 THEN A$=A$+MID$(STR$(F),2,3)
502 A$=A$+B$(B)
505 X=X(I)-X:Y=Y(I)-Y:A$=A$+"U"+MID$(STR$(ABS(X)),2,3)
510 C$="D":IF SGN(X)<-.5 THEN C$="H"
520 A$=A$+C$+MID$(STR$(ABS(Y)),2,3)
530 C$="F":IF SGN(Y)<-.5 THEN C$="B"
540 A$=A$+C$:X=X(I):Y=Y(I):F=1:B=5
550 A=0
590 GOTO 450
799 '******TO:RLE'S******
800 FOR I=1 TO F
810 X=X(K):Y=Y(K):RESET(X*2,Y):RESET(X*2+1,Y)
820 K=K-1:NEXT I
830 X=X(K):Y=Y(K):F=0
840 GOTO 200
900 FOR I=1 TO K:PRINTX(I);Y(I),:NEXT I
910 RETURN
```







SÍCSEL

Sokat vitatkoztunk barátaimmal, hogy lehet-e BASIC nyelven gyors játékprogramot írni. Ennek eldöntésére készítettem ezt a játékot. A program 1983 nyarán Zánkán készült az úttörőtáborban, az ott üdülő úttörők nagy örömére.
A program BASIC nyelven fut, de azért van benne egy kis csalafintaság is. Ez az ami miatt úgy érzem, helye van a BIT-LET-ben.

A képernyő egy sípályát ábrázol, az akadályokat a gép mozgatja, a sízőt pedig a játékos. A síző a képernyőn áll, a háttér mozog mögötte felfelé. Így az a látszat, hogy a síző jön lefelé a lejtőn. A sízőt a < gombbal balra, a > gombbal pedig jobbra lehet mozgatni. A pályát kétoldalt vonal zárja le, alulról pedig akadályok lépnek be, amelyeket ki kell kerülni. Akár a fallal, akár egy akadállyal történő ütközés hiba. Ekkor megáll a játék, és az ütközés helyét villogó pont jelzi. Rövid idő után a játék onnan folytatódik, ahol az ütközés történt. Összesen 9-szer lehet ütközni, ezután a játéknak vége van. Vége van akkor is, ha egy játékos 1000 „métert" tud kevesebb mint kilenc ütközéssel megtenni. Ekkor a gép gratulál a játékosnak. Ezt azért tettem bele a programba, mert alapvetően hibásnak tartom az olyan játékprogramot, ahol a játékos csak veszíthet. Ezzel elkerülhető, hogy a játékost a sikerélmény reményében újabb és újabb próbálkozásra sarkallja, és ne tudja abbahagyni a játékot.

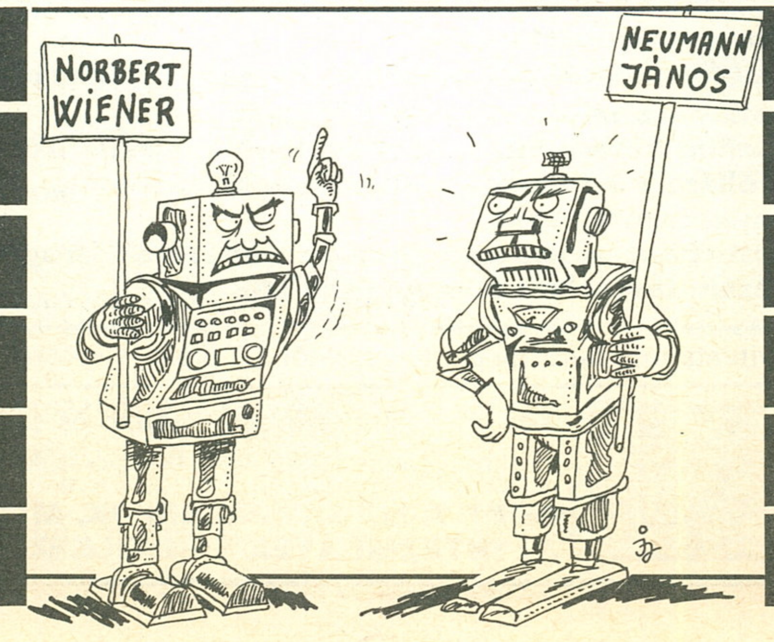
A programban sok olyan fogás van, amely gyorsítja a BASIC futását. Ezek a gyorsítások a következők:
- ahol csak lehet, egész számokat használjunk;
- a leggyakrabban használt változót a táblázat elejére teszi a BASIC, így a futás közben hamarabb találja meg;
- a PRINT TAB helyett használjuk a PRINT @ utasítást, ez sokkal gyorsabb;
- ne használjunk konstansokat, helyette tegyük be ezeket valamilyen változóba. (Ezt a fogást a program jobb áttekinthetősége érdekében nem alkalmaztam.).

Ezek a fogások sok BASIC-ben érvényesek. A következők csak a HT1080 Z és TRS 80 gépre igazak:
- annak érdekében, hogy a síző mozgatása és az akadályok mozgatása ne hátráltassa egymást, külön módszert alkalmaztam, s a képernyőt direkt módon kezelem. (Erről már jelent meg cikk a BIT-LET-ben.) Az akadályokat az a hatás mozgatja, hogy ha a képernyő utolsó pozíciójába írok, akkor a teljes képernyő egy sorral feljebb ugrik. Ezért lehetett gyors játékprogramot írni, hiszen így azt lehet mondani, hogy gépi kódú program mozgatja a figurákat. Ez a 90-es sorban történik.
- az INKEY$ rutin helyett a klaviatúrát is PEEK-utasítással érem el. (Erről is jelent meg már cikk.) Erről tehát nem kívánok semmi részleteset írni. Van valami, ami azonban még nem jelent meg egyik újságban sem. Ez pedig az, hogy a program sehol sem használja az eredeti BASIC klaviatúra rutint, tehát ki lehet tiltani. Ezzel jelentősen gyorsul a program, mivel a BASIC minden egyes utasítás után végignézi az összes billentyűt, s ez igen sok időt vesz igénybe. A kitiltást a következő utasítással lehet elérni: POKE 16406,82 FIGYELEM! Ezt az utasítást csak programból lehet kiadni, mert ezután már semmi utasítást nem fogad a BASIC a klaviatúrától, tehát a programot ~~sem~~ tudjuk elindítani. (Megjegyzem, a BREAK gombot sem fogadja el. Titkosítási lehetőség.) Ezzel ellentétes hatású a POKE 16406,227. Ezzel ismét működőképes a klaviatúra. Ezt a két utasítást célszerű a program elejére és végére beilleszteni. A program ezen kívül példát mutat arra is, hogyan lehet villogó karaktert vagy szöveget létrehozni.

Remélem sok tanulsággal szolgál ez a kis program, és kellemes perceket okoz azoknak, akik vállalják e néhány sor begépelését a gépbe. Kellemes időtöltést kíván:

Tráger Gábor

```
10 CLEAR:DEFINTA-Z:CLS
20 PRINT:PRINT"IDE IRHATO BE A HASZNALATI UTASITAS, ES A JATEKSZABALY"
30 IFPEEK(14432)<>1THEN30
40 CLS:P=15900:N=100:U=9:FORL=20TO120:FORR=0TO9
50 IFPEEK(14368)=64THENP=P+1:GOTO65
60 IFPEEK(14368)=16THENP=P-1
65 IFPEEKP<>32THENGOSUB500
70 POKEP,42:IFRND(N)<LTHENPRINT@960+RND(59),"####";
80 IFL=119THENPRINT@960,"CEL-CEL-CEL-CEL-CEL";:PRINT@1004,"CEL-CEL-CEL-CEL-CEL";:L=
L+1:R=4:N=32767
90 PRINT@1023,"!!";:NEXTR,L
100 FORI=0TO5:PRINT@535,"GRATULALOK, GYOZTEL !";:FORK=0TO80:NEXTK:PRINT@535,STRING$
(19," ");:FORK=0TO50:NEXTK,I:CLEAR:GOTO20
500 FORI=0TO5:POKEP,255:FORK=0TO50:NEXTK:POKEP,32:FORK=0TO50:NEXTK,I
510 IFP<=15872THENP=P+1
520 IFP>=15935THENP=P-1
530 U=U-1:IFU>0THENRETURN
540 PRINT@450,STRING$(192," "):PRIN@460,"TUL SOKAT UTKOZTEL, EZERT KIESTEL A JATEKBOL";
550 PRINT@532,"EDDIG"(L-20)*10"METERT TETTEL MEG":CLEAR:FORI=0TO1000:NEXTI:GOTO20
```



HT1080Z

TV DÖMPING

Egy iskolaszámítógépes kiállításon mutatta be nagy sikerrel Theisz György székesfehérvári tanár azt az egyszerű kis szerkentyűt, amely tanárok öröme lett országszerte. Ez a kis jószág lehetővé teszi, hogy a HT1080Z géphez szinte korlátlan mennyiségű tv-t csatlakoztassunk. Nosza, megkértük Theisz Györgyöt, hogy tegye közzé lapunkban, hogyan készíthető el ez a hasznos szerkezet.

A megoldás azon alapszik, hogy számítógépünk olyan hatalmas antennajelet biztosít a tv-készüléknek, hogy az büntetlenül leosztható, illetve csillapítható.

Ez tehát lehetővé teszi az antennajel passzív, ellenállásos hálózattal való elosztását. A mellékelt kapcsolási rajzról látható, hogy igen egyszerű, alkatrészt alig tartalmazó áramkörről van szó. Mindig ahány készüléket csatlakoztatunk az elosztó segítségével a számítógéphez, annál eggyel több ellenállásra van szükség, melyek értéke:

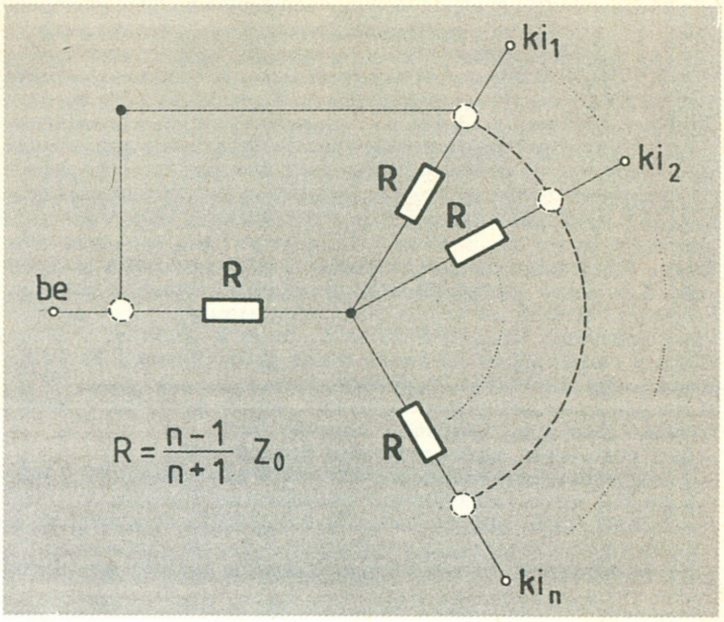
$$R = \frac{n-1}{n+1} Z,$$

ahol Z = 75 ohm. Mivel az illesztés nem túlságosan fontos hogy pontos legyen, így a gyakorlatban jól megfelelnek az alábbi értékek: 4 tv-ig R = 47 ohm, 5–8 között 56 ohm, efölött 68 ohm. Így a gyári adatok szerint a számítógép mintegy 30–100 (!) tv-készüléket képes antennajellel ellátni.

Megjegyzendő, hogy az elosztó használata a képminőséget némileg még javítja is, ugyanis a megfelelően (kb. 20 dB-lel) csillapított antennajel a tv-vevő optimálishoz közeli kivezérlését eredményezi.

Természetesen nem szükséges az összes kialakított kimenetet használnunk, ugyanis kevés készülék esetén az antennajel nagysága, sok készülék esetén a pontosabb illesztés biztosít megfelelő képet.

Az áramkör elkészítése igen egyszerű. Merev lapra (pl. fóliás lemezre) erősítsünk fel a vevők és a számítógép számára egy-egy koaxiális tv-antennacsatlakozó-aljat! Ezek hidegpontját forrasszuk csillagkapcsolásban! A csillagpontot ne csatlakoztassuk sehova! Az ellenállásokat minél rövidebb kivezetéssel szereljük, használjunk jó minőségű, axiális kivezetésű, kisméretű (legfeljebb negyedwattos) ellenállásokat! Az elosztó szerelésre egyébiránt nem nagyon kényes a viszonylag alacsony frekvencia miatt. A tv-készülékekhez szabványos csatlakozódugóval szerelt koaxiális kábellel csatlakozzunk!



LABIRINTUS

Szabó Gál András szentendrei másodikos gimnazista olvasónk küldte be a következő programot. Nem állítjuk, hogy problémátlan a program, de nagyon szívesen közöljük. Néhány ügyes programozási trükköt tartalmaz, s mint ilyen, „játékosan tanít". Jó tanács: vigyázzunk, hogy még véletlenül se menjen ki a bábu a képernyőről!

Sokan akik HT-ra írnak játékprogramot, leggyakrabban a SET és a RESET utasítást használják. Ennél van egy kényelmesebb és gyorsabb megoldás, a POKE. A 15360-as memóriacímtől kezdődik a képernyőmemória. Tehát ha azt írjuk be, hogy POKE 15360,143 akkor a képernyő bal felső sarkában megjelenik egy sötét pont [CHR $ (143)]. Az ütközéseket is könnyebben tudjuk ellenőrizni. Erre szolgál a PEEK (jelentése: bekukucskálni, benézni). Vagyis ha arra vagyunk kíváncsiak, hogy a képernyő 500. pozíciójában mi van, akkor csak azt kell beírnunk, hogy PRINT PEEK (15360+500) és akkor megtudjuk. A program ezt próbálja meg szemléltetni.

A játék maga az 5–16. sorig tart.

5. sor: 200 db pontot tesz ki véletlenszerűen

6. sor: a képernyő jobb alsó sarkába teszi ki a „kis házat" (ide kell majd eljutni)

7. sor: a PEEK (14350) kicsit hasonlít az INKEY $-ra, csak annyi a különbség, hogy az INKEY $-nál minden lépésnél le kell nyomni a gombot, így viszont nem. Hátránya viszont, hogy az IF-nél nem a betűt kell beírni, hanem annak a kódját (nem az ASCII-kódról van szó). Így az *i* 2-nek, a *j* 4-nek, az *m* 32-nek és a *k* 8-nak felel meg (a 2 hatványai).

8–11. sor megnézi, hogy mit nyomtunk le, a bábunk jelenlegi pozícióját eggyel odébbteszi, és a lépések számához is hozzáad egyet, majd csipog egyet.

14. sor: ellenőrzi, hogy nem mentünk-e neki semminek. De ha mégis ütközünk, akkor durrant, és az ütközések számát növeli a 18. sorban.

15. sor: arra ügyel, hogy ki ne menjünk a képernyőről, ellenkező esetben kellemetlen dolgok történhetnek. (Ki lehet próbálni a 15-ös sort IF nélkül.) Jó tanács: de csak azután, miután fölvettük magnóra, vagy kijátszottuk magunkat.

A program ennyi. A maradék már csak sallang.

A fontosabb változónevek:

CC	a display memória kezdete (15360)
A	az a 14350-es memória címértéke
X	a bábu pillanatnyi pozíciója a képernyőn
L	lépések száma
U	az ütközések száma

A játék lényege, hogy úgy kell átvergődnünk ezen a labirintuson, hogy 5 ütközésnél kevesebb legyen, és a bal felső sarokból átjussunk a jobb alsó sarokba.

```
1 REM SZABO GAL ANDRAS SZENTENDRE
2  CLS
3 PRINT:PRINT:PRINTTAB(25)"* HELLO *":PRINTTAB(16)"KERED A JATEK SZABALYOKAT?"
4 A$=INKEY$:IFA$="I" THEN GOSUB 23 ELSE IF A$="N" THEN 5 ELSE 4
5 CLEAR 1000:CLS:FORI=1 TO 200:POKERND(1023)+15360,127:NEXT:CC=15360
6 POKE 16381,187:POKE16379,183
7 LETA=PEEK(14350)
8 IFA=2THENPOKECC+X,128:X=X-64:L=L+1:GOSUB 22
9 IFA=32THENPOKECC+X,128:X=X+64:L=L+1:GOSUB 22
10 IFA=4THENPOKECC+X,128:X=X-1:L=L+1:GOSUB 22
11 IF A=8THEN POKECC+X,128:X=X+1:L=L+1:GOSUB 22
12 PRINT@50,L;" LEPES";
13 IF15360+X=16380THEN17
14 IF PEEK(15360+X)=127 THEN GOSUB 18
15 IF15360+X<16383THENPOKE15360+X,ASC("O")
16 GOTO 7
17 PRINT@8*64+10,"===``` GYOZELEM ```===";:GOSUB 31:FORI=1 TO20:PRINT@1019,CHR$(
183);"O";CHR$(187);:FORP=1TO10:NEXTP:PRINT@1019,"     ";:FORK=1TO17:NEXTK:NEXTI:G
OTO 19
18 U=U+1:PRINT@992,U;" UTKOZES";:IFU=5THENPRINT:PRINT"==** A JATEKNAK VEGE **=="
;:GOTO 19ELSEOUT 31,7:OUT30,7:FORH=15TO0STEP-1:FORG=0TO40:NEXTG:OUT31,8:OUT30,H:
OUT31,6:OUT30,31:NEXTH:RETURN
19 OUT31,7:OUT30,254:OUT31,8:OUT30,15:OUT31,0:FORI=1TO8:FORD=255TO1STEP-7:OUT30,
D:NEXTD,I
20 PRINT:PRINT@9*64+15,"UJ JATEK (I,N)?";
21 A$=INKEY$:IFA$="I"THENGOSUB34ELSEIFA$="N"THENCLS:PRINT@8*64+15,"=== VISZLAT =
==":OUT30,100:FORL=1TO10:NEXT:OUT30,0:ENDELSE 21
22 OUT 31,7:OUT30,248:OUT31,8:OUT30,15:OUT31,0:OUT 30,120:FORI=1TO4:NEXT:OUT30,0
:RETURN
23 CLS:A$="                          ** LABIRINTUS **":GOSUB30:PRINT:A$="ELTEVEDTEL
 EGY SURU ERDOBEN.HA KI AKARSZ JUTNI AZ ERDOBOL A KEP-":GOSUB30:A$="ERNYO JOBB
ALSO SARKABA-KELL ELJUTNOD A BABUDDAL 'O'.                ":GOSUB30
24 A$="LEHETOLEG MINNEL KEVESEBB UTKOZESSEL. (MAXIMUM 5)":GOSUB 30:PRINT:PRINT:A
$="====   ====   ====   ====   ====   ====   ====   ====   ==== ":GOSUB 30
25 A$="    IRANYITAS:    I
     J   K                                                        M  BILLE
NTYUK SEGITSEGEVEL TORTENIK":GOSUB 30
26 PRINT:A$="-----  J O  J A T E K O T -----":GOSUB 30:PRINT:A$="A PROGRAM 1984
FEBRUARJABAN KESZULT SZENTENDREN":GOSUB 30
27 PRINT:A$="NYOMJ LE EGY BILLENTYUT HA ELOLVASTAD !":GOSUB 30
28 IFINKEY$=""THEN28
29 GOTO 5
30 FORSZ=1 TO LEN(A$):PRINTMID$(A$,SZ,1);:FORT=1TO10:NEXTT,SZ:RETURN
31 DATA203,180,161,203,180,161,152,135,126,152,135,126,101,101
32 DIMH(14):FORI=1TO14:READH(I):NEXT
33 FORI=1TO14:OUT 30,H(I):FORT=1TO100:NEXTT:NEXTI:OUT 30,0:RETURN
34 A$=STRING$(63,191):B$=STRING$(63,128):PRINTCHR$(28):FORI=1 TO 16:PRINTA$:OUT3
0,I*6:NEXT:FORI=1TO16:OUT30,255-I*6:PRINTB$:NEXT:OUT 30,0:RUN 5
```

SZUPER BIT-LET

RÖVIDÍTÉSEK

a aritmetikai	*n* név
c constans	*s* string
e egyszerű	*u* utasításszám /0–65529/
i integer (egész)	*v* változó
k kifejezés	

α egy betű (A, B,...,Z)
γ egy tetsz. karakter
... rövidítés(ek)-nek megfelelő lista (elválasztójelként a vessző adható meg)
( ) utasításhoz tartozó zárójel, ki kell írni
[ ] a szögletes zárójelben levő paraméter használata nem kötelező
{ } a kapcsos zárójel egyik sorát kell választani
[̲ ̲ összetartozó utasítások
[CLEAR] kezelőgomb

MŰVELETEK

Aritmetikai:	+, –, *, /, ↑ (régi gépeken megjelenése: [)
Relációk:	<, <=, >, >=, =, <>
Logikai:	AND, OR, NOT Eredmény: –1, ha igaz a feltétel 0, ha nem igaz a feltétel A logikai műveleteket bitenként végzi a gép ($-1 \approx 1111\ 1111_2$)
String:	*sk*1 + *sk*2: összekapcsolás

MŰVELETEK SORRENDJE

Egy kategórián belül balról jobbra

1. zárójel
2. belső függvény
3. hatványozás
4. előjelváltás
5. szorzás, osztás
6. összeadás, kivonás
7. relációk
8. logikai műveletek a) NOT
   b) AND
   c) OR

VÁLTOZÓK TÍPUSÁNAK KIJELÖLÉSE

$i = 0,1,\ldots,9$

Változó (*vn*) alapján	Típusa	Pontossága	Definiálás alapján
$\alpha\left[\left\{ \begin{matrix} \alpha \\ i \end{matrix} \right\}\right]$ [!]	egyszeres pont.	max. 6 számj.	DEFSNG
$\alpha\left[\left\{ \begin{matrix} \alpha \\ i \end{matrix} \right\}\right]$ %	egész típusú	1	DEFINT
$\alpha\left[\left\{ \begin{matrix} \alpha \\ i \end{matrix} \right\}\right]$ #	dupla pont.	max. 16 számj.	DEFDBL
$\alpha\left[\left\{ \begin{matrix} \alpha \\ i \end{matrix} \right\}\right]$ s	karaktersorozat	–	DEFSTR

A név szerinti kijelölés a DEFααα típusú meghatározást felülírja

Értékterjedelmek:
egyszeres pont: –1.701411E+38 – 1.701411E+38
egész: –32768 – 32767
dupla pont: –1.701411834244556D+38 –
–+1.701411834244556D+38
karaktersorozat: 0 – 255 karakter

PARANCS ÜZEMMÓD

([NEW LINE]-ra végrehajtódik

NEW	tároló törlése
AUTO [*u*[,△*u*]]	automatikus sorszámozás
RE [*u*][,△*u*]	újraszámozza a programot – bővítésben
CLEAR	*i* byte-ot töröl string részére + törli a változókat
LIST [*u*1][–][*u*2]	program listázása
LLIST [*u*1][–][*u*2]	nyomtatóra listáz
DELETE [*u*1][–*u*2]	programsor(ok) törlése
EDIT [*u*]	áttérés szövegszerkesztő módba
RUN [*u*]	program futása (*u* sortól)
CONT	program folytatása
TRON	nyomkövetés bekapcsolása
TROFF	nyomkövetés kikapcsolása
CSAVE $\left[\#\text{-}\left\{ \begin{matrix} 1 \\ 2 \end{matrix} \right\},\right]$ *sc*	BASIC program kazettára írása
CLOAD $\left[\#\text{-}\left\{ \begin{matrix} 1 \\ 2 \end{matrix} \right\},\right]$?[*sc*]	a kazettán és a memóriában levő program összehasonlítása
CLOAD $\left[\#\text{-}\left\{ \begin{matrix} 1 \\ 2 \end{matrix} \right\},\right]$ [*sc*]	BASIC program olvasása kazettáról
SYSTEM	monitor üzemmódba áttérés

Minden parancs (a CONT kivételével) használható utasításként is. Utasításszám helyére mindenhova írhatunk egy pontot is, mely az aktuális sor (az utoljára beírt, javított, ill. listázott sor) sorszámát helyettesíti.

MONITOR ÜZEMMÓD

i,*i1*,*i2* = 4000,...,7FFF (hexadecimális)

/12710	monitor aktivizálása + CPU regisztereinek aktuális tartalma
R	CPU regiszterek módosítása (X továbbugrás)
M*i*	memória módosítása (*i*.-től, X kiugrás)
D*i*	memóriatartalom megjelenítése (*i*.-től X vissza)
B	visszatérés BASIC-be
G *i1*[,*i2*]	program futtatása *i1* címtől *i2* a töréspont
/12294	kisbetű ([SHIFT]-tel)
/12299	– – – „ – – –, bill.ism., RE parancs
/12288	– – – „ – – –, – – – „ – – –, – – – „ – – –, vill. cursor

SZÖVEGSZERKESZTŐ ÜZEMMÓD

EDIT $\left\{ \begin{matrix} . \\ u \end{matrix} \right\}$	aktuális, ill. *u*-adik javítható
[NEW LINE]	tárol minden változtatást
[*i*] [SPACE]	kurzor *i*-vel jobbra vitele
[*i*] [←]	kurzor *i*-vel balra vitele
[SHIFT] [↑]	befejezi a közbeírási parancsot
H	jobbra eső rész törlése + beszúrás
I	beszúrás (jobbra tolással)
X	a sor végére állás + beszúrás
L	sor kilistázása
A	töröl mindent változtatást
E	tárolja a változtatásokat
Q	vissza a parancs üzemmódba a változások tárolása nélkül
[*i*]D	*i* karaktert töröl
[*i*]C	*i* karaktert cserél
[*i*]Sγ	*i*-edik γ karakter elé állás
[*i*]Kγ	*i*-edik γ karakterig töröl

Ha *i* hiányzik, akkor értéke 1.

HT PUSKA

BELSŐ ARITMETIKAI FÜGGVÉNYEK

$x=ak$, szögek esetén radián

Átszámítások: rad←fok·π/180; fok←rad·180/π

$\lg(x) \leftarrow \ln(x)/\ln(10); 10^x = e^{x\cdot\ln(10)}$

π=3,14159265358979 e=2,71828182845905

SIN (*x*)	sin *x*	LOG (*x*)	ln *x*
COS (*x*)	cos *x*	EXP (*x*)	e^x
TAN (*x*)	tg *x*	ABS (*x*)	|*x*|
ATN (*x*)	arc tg *x*	SQR (*x*)	$\sqrt{x}$

FIX (*x*)	csonkít
INT (*x*)	*x* egész részét adja (az eredmény valós)
CINT (*x*)	*x* egész részét adja (az eredmény egész típusú)
CSNG (*x*)	egyszeres pontosságúvá alakít
CDBL (*x*)	kétszeres pontosságúvá alakít
SGN (*x*)	előjel függvény: −1, ha *x*<0; 0, ha *x*=0; 1, ha *x*>0
RND (*x*)	*x*=0: egyszeres pontosságú véletlenszámot ad a (0, 1) intervallumban *x*=*i*: 1 és *i* közötti egész számot állít elő

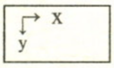
STRINGEKKEL KAPCSOLATOS FÜGGVÉNYEK

ASC(*sk*)	első karakter ASCII kódját adja
CHR$(*sk*)	ASCII kód alapján karaktert ad (0≤*ak*<255)
LEN(*sk*)	string hosszát adja
LEFT$(*sk, ak*)	string bal oldali része (*ak* db.)
MID$(*sk, ak1, ak2*)	string közepéből vág ki (az *ak1*. karaktertől kezdve *ak2* db.-ot)
MID$(*sk,ak*)	string jobb oldali része (*ak*.-tól)
RIGHT$(*sk, ak*)	string jobb oldali része (*ak* db.)
STRING$ (i, {*sc* / ASCII kód})	adott karakterből *i* hosszúságú stringet állít elő (*i*=0,...,255)
STR$(*ak*)	stringgé alakít
VAL(*sk*)	numerikus értékké alakít

GRAFIKUS FUNKCIÓK

x=*ak1*, oszlopok száma 0,...,127

y=*ak2*, sorok száma 0,...,47



CLS	a képernyő tartalmának törlése
SET(*x, y*)	fehér téglalap kerül az *x,y* helyére
RESET(*x, y*)	a fehér téglalapot törli az *x,y* helyen
POINT(*x, y*)	a függvény értéke −1, ha az *x,y* helyen van fehér téglalap; 0, ha nincs

HIBAÜZENETEK

1 NF FOR nélküli NEXT	13 TM típuskeverés
2 SN szintaktikai hiba	14 OS stringterület kicsi
3 RG GOSUB nélküli RETURN	15 LS a string túl hosszú
4 OD nincs több adat	16 ST túl összetett stringműv.
5 FC hibás függvényutasítás	17 CN CONT-tal nem folytatható
6 OV túlcsordulás	
7 OM nincs több tárolóhely	18 NR hiányzó RESUME
8 UL nem létező sor	19 RW RESUME, hiba nélkül
9 BS indextúllépés	20 UE nem kiírható hiba
10 DD már dimenzionált	21 MO operandus hiányzik
11 /0 nullával osztás	22 FD hiányos file
12 ID hibás felhasználás	

BASIC-UTASÍTÁSOK

ÉRTÉKADÁS — TÖMB MÉRETEZÉSE

[LET] {*av*=*ak* / *sv*=*sk*}   DIM *vn*(*i1*[,*i2*,...,*in*])

PRINT [TAB (*ak*)][*k*][{,;}]... képernyőre írás

RANDOM   az RND (*x*) függvény véletlenszámtól kezdődik

BE- ÉS KIÍRATÁSI UTASÍTÁSOK

LPRINT	sornyomtatóval ír
PRINT @ *i,k*...	*i*. képernyőpozicióra íratás ha *i1*. sor (0≤*i1*≤15) *i2* oszlopába (0≤*i2*≤63) akarunk íratni, akkor *i*=64·*i1*+*i2* (*i*=0,...,1023)
PRINT [@*i*,] USING *sk*...	kiíratás formátum szerint
INPUT [*sc*;]*v*...	adatbevitel billentyűzetről
┌ DATA *c*...	adatok tárolása a pr.-ban
│ READ *v*...	olvasás a DATA-ból
└ RESTORE	DATA elejére állás
PRINT # − {1 / 2},k...	kazettára írás
INPUT # − {1 / 2},v...	kazettáról beolvasás
REM	megjegyz. elhely. a pr.-ban

A PRINT kulcsszó ?-lel, a REM '-fal rövidíthető.

VÉGREHAJTÁS VEZÉRLÉSE

GOTO *u*	ugró utasítás
ON *ak* GOTO *u1, u2*,...	elágazó ugró utasítás
┌ GOSUB *u*	alprogramra ugrás
│ ON *ak* GOSUB *u1,u2*,...	elágazó alprogr. ugrás
└ RETURN	alprogr. vége, visszatérés
IF *k* [[THEN]{*u1* / utasítás(ok)}] [ELSE{*u2* / utasítás(ok)}]	feltételes utasítás
┌ FOR *aev*=*ak1* TO *ak2* [STEP *ak3*] └ NEXT [*aev*]...	ciklusszervezés
┌ ON ERROR GOTO *u*	hiba esetén ugrik, ha *u*>0
└ RESUME [{*u* / NEXT}]	pr. végrehajtás folytatása
ERROR *i*	*i* kódú hiba szimulálása
STOP	programfutás megszakítása
END	programfutás megszakítása

Minden utasítás – az INPUT és az INPUT# kivételével használható parancsként is.

SPECIÁLIS FUNKCIÓK

i=0,...,32767

i1=15360,...,32767

i2=0,...,255

PEEK(*i*)	*i*-edik byte értékét adja
POKE*i1*, *i2*	*i1*-edik byte-ba *i2* értékét írja
FRE(*s*)	stringek számára szabad byte-ok száma
FRE(*av*)	memória szabad byte-jainak száma
MEM	memória szabad byte-jainak száma
ERL	hibás sor számát tartalmazza
ERR	ERR=(hibakód−1)*2
INKEY$	utolsó INKEY$, ill. INPUT után legutoljára leütött karakter, ha ilyen nincs: ""
VARPTR(*evn*)	változó értékének kezdőcímét adja a memóriában

SZUPER HT PUSKA

SPECIÁLIS FUNKCIÓK

INP (*i*)	az *i*. bemenetről olvas egy byte-ot *i*=0,...,255
OUT *i1*,*i2*	az *i1*. kimenetre kiírja *i2*-t *i1*,*i2*=0,...,255
POS (*k*)	a kurzor aktuális helye
USR (*k*)	meghívja a 256*(16527)+(16526) címen kezdődő gépi rutint

SPECIÁLIS BILLENTYŰK, GOMBOK

[RESET]	a gép alapállapotba kerül, a memória nem törlődik
[VIDEO CUT]	váltás széles karakterekre és vissza
[PAGE]	széles karakter esetén váltás a képernyő bal, ill. jobb felére
[F1]	áram a magnónak
[BREAK]	programvégrehajtás megszakítása
[SHIFT]/[BREAK]	————„———— + a kurzor villogásának be-, ill. kikapcsolása
[CLEAR]	képernyő tartalmának törlése + visszaállás sűrű (64 kar./sor) írásmódra (új gépeknél, ha a bőv. be van kapcs.: [CLEAR]/[BREAK])
[NEW LINE]	aktuális sor, ill. adatbevitel vége
[←]	1 karakterrel visszalép és töröl
[SHIFT]/[←]	aktuális sor törlése
[→]	kurzort a következő nyolcad elejére állítja
[SHIFT]/[→]	átállás ritka (32 kar./sor) írásmódra
[↓]	kurzort a következő sor elejére viszi
[SHIFT]/[↓]	a köv. billentyű kódját 64-gyel csökkenti
[SHIFT]/[@]	futás, ill. listázás felfüggesztése (folytatás: tetsz. bill., kiv. [BREAK])
[SPACE]	betűköz

Új gépnél, ha a bővítés be van híva:

[CLEAR]/[betű]	A,E,O,U,a,e,o,u betűből Á,É,Ö,Ü,á,é,ö,ü betűt csinál

NÉHÁNY ASCII KÓD

8	[←] kódja, 1 karakterrel visszalép és töröl
9	[→] kódja
10	soremelés
13	[NEW LINE] kódja, új sort kezd
23	váltás ritkított írásmódra
24	1 karakterrel visszalép
25	1 karakterrel előre lép
26	1 sorral lejjebb lép
27	1 sorral feljebb lép
28	kurzor a bal felső sarokba + sűrű írásmód
29	kurzor az aktuális sor elejére
30	törli a kurzortól jobbra eső sorrészt
31	törli a kurzor mögötti képernyőrészt
32	szóköz
34	"
48	Ø
65	A
97	a
127	pepita karakter
128–191	grafikus kódok

kód-128-at kettes számrendszerbe írva megkapjuk, hogy melyik kis téglalap világít:

1	2
4	8
16	32

192–255	kód-192 db. betűköz

NÉHÁNY CÍM

HEX.		DEC.
7FFF	RAM	32767
42E9		17129
42E8	BASIC rendszervált.	17128
4000		16384
3FFF	Képernyő	16383
3C00		15360

HEX.		DEC.
3BFF	Billentyűzet	15359
3800		14336
35FF	EPROM	13823
3000		12288
2FFF	ROM	12287
0000		0

16544–16545	BASIC stringterület kezdőcíme
16548–16549	BASIC program kezdőcíme (17129)
16551–16552	a billentyűzet-puffer kezdőcíme
16559	aktuális változó típusjelzője (2 – egész, 3 – string, 4 – valós, 8 – dupla pont.)
16561–16562	BASIC munkaterület teteje
16607–16608	SYSTEM-ben betöltött gépi kódú program kezdőcíme
16633–16634	skalártáblázat kezdőcíme
16635–16636	vektortáblázat kezdőcíme
16637–16638	BASIC feletti szabad tár kezdőcíme

Billentyűzet:

BIT	TÁRCÍM							
	14337	14338	14340	14344	14352	14368	14400	14464
0	@ `	H h	P p	X x	ø	8 (	NEW LINE	SHIFT
1	A a	I i	Q q	Y y	1 !	9)	CLEAR	
2	B b	J j	R r	Z z	2 "	: *	BREAK	
3	C c	K k	S s		3 #	; +	↑	
4	D d	L l	T t		4 $	, <	↓	
5	E e	M m	U u		5 %	– =	←	
6	F f	N n	V v		6 &	. >	→	
7	G g	O o	W w		7 '	/ ?	SPACE	

Hang:

Szabályozás: OUT 31,*i1* : OUT 30,*i2*
az *i1*. regisztert *i2*-re állítja

Regiszterek:
0 – „A" csat. hangmagasság finom szab. (0-255)
1 – „A" csat. hangmagasság durva szab. (0-15)
2 – „B" csat. hangmagasság finom szab. (0-255)
3 – „B" csat. hangmagasság durva szab. (0-15)
4 – „C" csat. hangmagasság finom szab. (0-255)
5 – „C" csat. hangmagasság durva szab. (0-15)
6 – zajgenerátor (0-31)
7 – keverő: ha a Ø.,1.,2. bit Ø:„A",„B",„C" csat. eng.
ha a 3.,4.,5. bit Ø:„A",„B",„C"-re zaj kev.
a 6. és 7. bitet állítsuk 1-re!
8,9,1Ø – „A",„B",„C" hangerő szab. (Ø-15, 16:burk. g. alapj.)
11,12 – a burkológörbe generátor finom és durva szab. (0-255)
13 – burkológörbe hullámformájának kiválasztása (0-15)

SORTÖRLÉS

Az ABC 80 számítógépen a BASIC forrássorok csak egyesével törölhetők. Hosszabb programrészek törlésekor fárasztó a sorszámok begépelgetése. A mellékelten közölt programmal két sorszám közötti BASIC sorok egyszerre „kigyilkolhatók". A program kis mértékben függ attól, hogy milyen ellenőrző összegű ABC 80-on akarják futtatni. Az ellenőrző összeg a BASIC interpreter byte-jainak az összege, és az ABC 80 alapegységen, a billentyűzet alatt található meg. Pl. CHEKSUM: 11273

Az alábbi program a 11273 ellenőrző összegű ABC 80 gépeken futtatható, a CHECKSUM 10042 ellenőrző összegű gépeken a programban aláhúzott byte-ok helyére 1CH-t kell beírni.

Azt, hogy ettől eltérő ellenőrző összegű gépeknél hogyan lehet megkeresni a beírandó byte-ok értékét, a működési leírás tartalmazza.

A programot a beírás és az adott gépnek megfelelő módosítás elvégzése után célszerű kazettára elmenteni. A programot a RUN utasítással egyszer le kell futtatni. A sortörléshez szükséges Z80 gépi kód az ABC 80 tárának legfelső 128 byte-jára kerül. Ezután az ABC 80 kikapcsolásáig a sortörlő rutin bármikor használható, a

; CALL (—128)

utasítással. (Ez egyenértékű a CALL (65408) utasítással, csak 1 karakterrel rövidebb, és könnyebben megjegyezhető.)

A rutin kiírja, hogy ő a sortörlő, majd két szám begépelését kéri:

LINE DELETE

TYPE # 1, # 2:

Válaszul a törlendő tartomány határain lévő sorszámokat kell megadni. Ha valamelyik sorszám nem található meg a programban, akkor ERR 6 hibajelzést ad a rutin. Túl nagy, vagy hibás sorszámok hatására ERR 31-es hibajelzés keletkezik. Ha helyesen történt a sorok megadása, akkor egy pillanat alatt lezajlik a sorok törlése, és a BASIC interpreter visszakapja a vezérlést.

```
ABCD
100 REM ===LINE DELETE LOADER===
110 DATA 21,D8,FF,01,18,00,CD,0B
120 DATA 00,21,40,FE,E5,01,78,00
130 DATA CD,05,00,E1,E7,CD,1E,18
140 DATA 38,58,D5,7E,FE,2C,20,52
150 DATA DF,CD,1E,18,38,4C,7E,FE
160 DATA 0D,20,47,EB,D1,E5,A7,ED
170 DATA 52,38,3F,CD,39,0F,20,38
180 DATA E3,EB,CD,39,0F,20,31,5E
190 DATA 16,00,19,E5,EB,2A,1E,FE
200 DATA 23,A7,ED,52,44,4D,E1,D1
210 DATA ED,B0,1B,ED,53,1E,FE,C9
220 DATA 4C,49,4E,45,20,44,45,4C
230 DATA 45,54,45,0D,0A,54,59,50
240 DATA 45,20,23,31,2C,23,32,3A
250 DATA D7,86,D7,9F,VEGE
260 A%=65408% : H¤='0123456789ABCDEF'
270 READ Q¤ : IF Q¤='VEGE' THEN 300
280 POKE A%,(INSTR(1%,H¤,LEFT¤(Q¤,1%))
-1%)*16%+INSTR(1%,H¤,RIGHT¤(Q¤,2%))-1%
290 A%=A%+1% : GOTO 270
300 END
```

„Ha a táj meg a térkép

nem egyezik meg,

akkor a táj szerint kell eljárni!"

(Svéd katonai közmondás)

HOGY MŰKÖDIK?

Azok számára, akik nem csak használni akarják a sortörlő rutint, hanem a működése is érdekli őket, a 2.ábra tartalmazza a lefordított program listáját. A fordítás ABC 80-on készült. A szabványos formától való egyetlen eltérés az, hogy az "EQU" direktívában szereplő szimbólumok után áll " : ".

A gépi rutin két részből áll: az első rész a szövegkiírást és a sorok bevitelét végzi, a második rész pedig a tényleges törlést. Mindkét részben az ABC 80 BASIC interpreterben szereplő szubrutinok hívására is sor kerül. Először ezeknek a szubrutinoknak a leírását adjuk meg:

000BH, WRITE szövegkiíró szubrutin
- **INPUT:**
  - HL a bemeneti puffer kezdetére mutat
  - BC a megengedett maximális sorhosszúságot tartalmazza
- **OUTPUT:**
  - Az input pufferbe beolvasott karakterek, amelyeket CR (0DH) zár.

181 EH, ASC12 számkonvertáló szubrutin
- **INPUT:**
  - HL a szöveg kezdetének címe,
  - BC a szöveg hossza.

0005H INPUTL, sorbaolvasó szubrutin
- **INPUT:**
  - HL a bemeneti puffer kezdetére mutat,
  - BC a megengedett maximális sorhosszúságot tartalmazza.

```
                   ;ADDRESS OF ABC80 ROUTINES
000B=              WRITE: EQU  0BH      ;WRITE TEXT
0005=              INPUTL:EQU  05H      ;INPUT LINE
0F39=              FNDLIN:EQU  0F39H    ;FIND LINE
181E=              ASCI2: EQU  181EH    ;CHARO INT CO

                   ;ADDRESS OF ABC80 SYSTEM VARIABLES
FE1C=              BOFA:  EQU  0FE1CH   ;BEGINNING OF
                   ;                     BASIC ADDR.
FE1E=              EOFP:  EQU  0FE1EH   ;END POINTER OF
                   ;                     BASIC PROGRAM
FE40=              INPB:  EQU  0FE40H   ;INPUT LINE BUFF

                   ;EQUATES
000D=              CR:    EQU  0DH      ;CURSOR RETURN
000A=              LF:    EQU  0AH      ;LINE FEED
                   ;-----------
                   ;INPUT PART
                   ;-----------
                   ;READ LINE#1 & LINE#2 TO REGISTERS
                   ;DE & HL

                   ;PRINT PROMPT
FF80  21D8FF              LD   HL,MSG1 ;MESSAGE ST.ADDR
FF83  011800              LD   BC,MSG1E-MSG1 ;LENGTH
FF86  CD0B00              CALL WRITE

                   ;INPUT ONE LINE
FF89  2140FE              LD   HL,INPB ;BUFFER ST. ADD.
FF8C  E5                  PUSH HL      ;SAVE IT
FF8D  017800              LD   BC,120  ;MLINE LENGTH
FF90  CD0500              CALL INPUTL  ;INPUT LINE
FF93  E1                  POP  HL      ;RESTORE BUFF S.

                   ;READ IN LINE#1 & LINE#2
FF94  E7                  RST  20H     ;READ A NONSPACE
FF95  CD1E18              CALL ASCI2   ;READ I2 TO DE
FF98  3858                JR   C,ERR31 ;OVERFLOW
FF9A  D5                  PUSH DE      ;SAVE LINE#1

FF9B  7E                  LD   A,(HL)  ;CHECK DELIMITER
FF9C  FE2C                CP   ','     ;COMMA?
FF9E  2052                JR   NZ,ERR31 ;NO:ERROR

FFA0  DF                  RST  18H     ;YES:READ NEXT
                                       ;NONSPACE CHAR

FFA1  CD1E18              CALL ASCI2   ;READ I2 TO DE
FFA4  384C                JR   C,ERR31 ;OVERFLOW
FFA6  7E                  LD   A,(HL)  ;CHECK TERMINATR
FFA7  FE0D                CP   CR      ;CURSOR RETURN?
FFA9  2047                JR   NZ,ERR31;NO:ERROR

FFAB  EB                  EX   DE,HL   ;HL=LINE#2
FFAC  D1                  POP  DE      ;DE=LINE#1
```

OUTPUT:
Az input pufferbe beolvasott karakterek, amelyeket CR (0DH) zár.

181EH, ASC12 számkonvertáló szubrutin
INPUT:
HL a számot ASCII karakterek formájában tartalmazó puffer címe.
OUTPUT:
CY hibajelző flag, akkor billen be, ha a szám érvénytelen (pl. „VVV") vagy túlságosan nagy (pl. „111111").
Hiba esetén HL értéke változatlan.
DE a beolvasott 2 byte-os egész szám.
Helyes számbeolvasás esetén a HL a számot határoló karakterre mutat (pl. „123," esetén a ","-re).

0F39H, FNDLIN BASIC sorszámkereső szubrutin
INPUT: DE a keresett sorszám.
OUTPUT:
Z értéke igaz, ha keresett sort megtalálta,
HL a keresett sorszámú sor kezdőcímét tartalmazza.

0020H, RST 20H karakterolvasás szóközök elhanyagolásával
INPUT: HL puffermutató.
OUTPUT:
HL a szóköztől eltérő karakterre mutat,
A a szóköztől eltérő karaktert tartalmazza.
Flag-ek az A-nak megfelelően beállítva

0018H, RST 18H karakterolvasás, szóközök elhanyagolásával, a puffermutató előzetes növelésével.
I/O: lásd RST 20H.

```
                ;---------------
                ;PROCESSING PART
                ;---------------
                ;DELETE LINES LYING BETWEEN LINE#1
                ;(REG DE) & LINE#2 (REG HL)

                ;CHECK LINE#'S
FFAD  E5                PUSH HL       ;          (1)
FFAE  A7                AND  A
FFAF  ED52              SBC  HL,DE    ;LINE#2<LINE#1?
FFB1  383F              JR   C,ERR31  ;YES:ERROR

FFB3  CD390F            CALL FNDLIN   ;LINE#1 FOUND?
FFB6  2038              JR   NZ,ERR6  ;NO:ERROR
FFB8  E3                EX   (SP),HL  ;SV LIN1 ADDR
FFB9  EB                EX   DE,HL
FFBA  CD390F            CALL FNDLIN   ;LINE#2 FOUND?
FFBD  2031              JR   NZ,ERR6  ;NO:ERROR

                ;COMPUTE NEXT LINE ADDRESS
FFBF  5E                LD   E,(HL)
FFC0  1600              LD   D,0
FFC2  19                ADD  HL,DE    ;ADD DISPL.
FFC3  E5                PUSH HL       ;          (2)

                ;MOVE LAST PART OF PROGRAM
FFC4  EB                EX   DE,HL
FFC5  2A1EFE            LD   HL,(EOFP)
FFC8  23                INC  HL       ;01 MUST BE
                                      ;MOVED TOO
FFC9  A7                AND  A
FFCA  ED52              SBC  HL,DE    ;LEN OF LASTPART
FFCC  44                LD   B,H
FFCD  4D                LD   C,L
FFCE  E1                POP  HL       ;NEXT LINADD(1)
FFCF  D1                POP  DE       ;LIN1 ADDR  (0)
FFD0  EDB0              LDIR
FFD2  1B                DEC  DE
FFD3  ED531EFE          LD   (EOFP),DE ;POINTS TO 01
FFD7  C9                RET

FFD8  4C494E    MSG1:   DEFM 'LINE DELETE'
      452044
      454C45
      5445
FFE3  0D                DEFB CR
FFE4  0A                DEFB LF
FFE5  545950            DEFM 'TYPE #1,#2:'
      452023
      312C23
      323A
FFF0=           MSG1E:  EQU  $

FFF0  D7        ERR6:   RST  10H
FFF1  86                DEFB 80H+6

FFF2  D7        ERR31:  RST  10H
FFF3  9F                DEFB 80H+31

                END
```

0010H, RST 10H hibajelző szubrutin.
Az RST 10H után szereplő byte-ban 80H+A kijelzendő hiba számának kell állnia. A hibajelzést a BASIC interpreter végzi.
És ez még nem minden! A sortörlő a fenti szubrutinokon kívül még két rendszerváltozót és egy rendszerpuffert is használ:
BOFA a BASIC program kezdőcímét tartalmazó szó (két byte),
EOFP a BASIC program végére mutató szó.
INPB az INPUTLINE BASIC utasításban használt bemeneti puffer.

Ennyi előzetes után rátérhetünk a program ismertetésére. Az első rész az ún. „INPUT PART", a fenti szubrutinok használatával a HL és a DE regiszterekbe teszi a két beolvasott számot. Érdekesebb a második rész, az ún. „PROCESSING PART". Itt történik meg az input adatok helyességének a vizsgálata (ti. LINE1 ≦ LINE2 fennáll-e) ,illetve annak a vizsgálata, hogy léteznek-e a BASIC programban a megadott sorszámú sorok. Ha a sorok léteznek, akkor a sortörlő rutin megkeresi a LINE2 után következő BASIC sor címét. Az ABC 80-nál ez igen egyszerűen történhet, ugyanis a BASIC sorok szerkezete a következő:

a_1	b_1	c_1	d_1	a_2	b_2	c_2	d_2	. . .

ahol a_1 a sor hossza (1 byte), b_1 c_1 a sor sorszáma (két byte) d_1 a BASIC sor tartalma. Így tehát a LINE2 kezdőcíméhez a LINE2 hosszát hozzáadva megkapható a következő sor címe is. Ezek után már csak a LINE2-t követő sortól az EOFP által mutatott byte-ig terjedő területen lévő byte-okat kell átmozgatni a LINE1 által mutatott területre, és az új programnévnek megfelelően fel kell újítani az EOFP tartalmat.

A VÁLTOZÓ BYTE-OK

A leírás első részében említettük, hogy eltérő ellenőrző öszszegű gépeknél megváltozhat a program. Ennek az az oka, hogy az FNDLIN és az ASCI2 rutinok az interpreter változattól függően más-más címen kezdődhetnek. Az alábbiakban megadjuk, hogy a rutinok milyen byte-okkal kezdődnek:
0F39H (3897) FNDLIN, 2AH, 1CH, FEH
181EH (6174) ASCI2, 7EH, 11H, 00H

A SZÜKSÉGES ELJÁRÁS:

1. A PEEK utasítással megkeresni a szubrutinok kezdőcímét.
2. A kezdőcím alsó byte-ját hexadecimális számmá kell alakítani.
3. Az így kapott két hexadecimális számjegyet az aláhúzott byte-ok helyett kell beírni.

Közreadta: ***Liposzky Gábor***



SZUPER BIT-LET

KARAKTERCSERE

A Primo számítógép számos – eddig még nem ismertetett – tulajdonsággal rendekezik. Ezek közül az alábbiakban egy olyan lehetőséggel foglalkozunk részletesen, amely minden bizonnyal sok Primo-felhasználó érdeklődésére tarthat számot.

Aki már kipróbálta a Primot, tapasztalhatta, hogy a képernyőn a normál karaktereken túl aláhúzott, invertált és nyújtott – dupla szélességű – karakterek is megjeleníthetők. Lehetséges még alsó, ill. felső indexbe írni, sőt a Primo még arra is képes, hogy a karakterek az óramutató járásával ellentétes irányban 90°-kal elforgatva jelenjenek meg. Új jeleket is létrehozhatunk úgy, hogy több karaktert egymásra írunk. (Pl. az = és a / egymásra írásával megjeleníthető a ≠ karakter). A Primo az első olyan hazai személyi számítógép, amely képes a magyar nyelv valamennyi ékezetes kisbetűjének és az ékezetes nagybetűk egy részének megjelenítésére.

A felsorolt számos lehetőségen túl a Primo-képernyő kezelése azt is megengedi, hogy a felhasználó új, a gép által még ismeretlen karaktereket hozzon létre, melyek definiálásuk után a szabványos jelekhez hasonlóan alkalmazhatók. Mivel 128 különféle karakter definiálására van mód, így pl. megoldható, hogy a Primo még hiányzó ékezetes nagybetűket, cirill betűs szövegeket vagy görög betűket alkalmazó képleteket, esetleg valamely játékprogramban szükséges különleges figurákat jelenítsen meg. Annak érdekében, hogy e lehetőséget kihasználhassuk, vizsgáljuk meg részletesebben, hogy is jelenik meg egy karakter a számítógéphez kapcsolt tv képernyőjén.

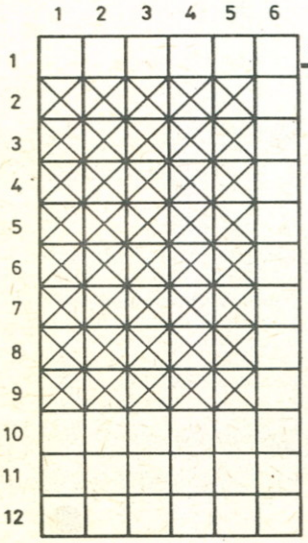
A Primo képernyője 256x192 pontból álló négyzetháló. A 49 152 pont bármelyike a többitől függetlenül világosra, ill. sötétre változtatható. A pontokból összeállított kép megjelenítését a Primo megfelelő áramkörei végzik. A pontraszter megjelenítésén túl a hardver semmilyen segítséget nem nyújt a működtető program számára. Nincs karakter – RAM, nincs hagyományos karaktergenerátor – ROM. A képernyőn megjelenő karaktereket a működtető program pontonként rajzolja be a négyzethálóba. Ez a megoldás ugyan jelentősen megnöveli a képernyőkezelő program feladatait, de lehetővé teszi a Primo képernyő áramköreinek egyszerűsítését, valamint a bevezetőben már említett számos különleges lehetőség egyszerű megvalósítását.

A Primo normál méretű karakterei vízszintesen 5, függőlegesen 8 pont kiterjedésűek. Az 5x8 pont egy 6x12 pontból álló mezőben helyezkedik el, így egyidőben 16 sorban, soronként 42 karakter jeleníthető meg. A karakter a rendelkezésre álló mezőben az 1. ábrán látható módon helyezkedik el. Mint látható, a karakter jobb oldalán elhelyezkedő 6. oszlop üresen áll, ez biztosítja a szomszédos karakterek elválasztását. Az 1. és 10–12. sorok az egymás alatt, ill. fölött álló karakterek közti üres területet jelképezik. Amikor a képernyőkezelő program egy karaktert a képernyőre rajzol, kikeresi a Primo ROM-jában elhelyezkedő karaktergenerátor területről a megfelelő karakter pontmintázatát. Ez egy karakterenként 8 byteos terület, amely meghatározza, hogy mely pontokat kell kigyújtani, ill. eloltani annak érdekében, hogy a képernyőn a kívánt betű, szám vagy írásjel megjelenjen. A pontmintázat minden byte-ja a karakter egy sorát határozza meg. Mint említettük, egy karaktermező 6 pont széles és 12 pont magas. Egy sor meghatározásához a byte 6 bitje szükséges tehát. Az 1. oszlopot a 6-os bit, a 2. oszlopot az 5-ös bit, míg az 5. oszlopot az 1-es bit jelképezi. (A byte bal szélső bitje a 7-es, jobb szélső bitje a 0-ás sorszámú). A megmaradt két bit – 7-es és 0-ás – speciális célokat szolgál.

Ha az 1. sort meghatározó byte 7-es bitjében 1 áll, a karakter ún. süllyesztett betű. Ilyen pl. az y vagy a j. Az ilyen karaktereket a képernyőkezelő program a 6x12 pontból álló mezőben nem a 2. sortól, hanem a 4. sortól kezdve jeleníti meg. Ennek megfelelően a karakter legalsó sora a mező 11. sorába kerül. Ezzel a megoldással elértük, hogy egy nyomtatott vagy írott szöveghez hasonlóan a süllyesztett karakterek egyes részei az írásvonal alá nyúlva jelenhetnek meg, de ezek a karakterek nem igényelnek 8 byte-nál hosszabb pontmintázatot. A karakter további sorait meghatározó byte-ok 7-es bitjének értéke közömbös.

A 7-es bithez hasonlóan a 0-ás bitet is speciálisan értelmezi a képernyőkezelő program. Ezzel a bittel jelezzük a program számára, hogy kirajzolta már a karakter legalsó sorát, nem kell újabb sort meghatározó byte-ot kiolvasni a karaktergenerátor területéről. Ez a megoldás – amely lehetővé tette a képernyőkezelő program egyszerűsítését – azt a további előnyt is nyújtja, hogy 8 pontnál magasabb karaktereket is alkalmazhatunk. Így pl. több soron átnyúló integráljel is megjeleníthető. Ez úgy oldható meg, hogy a karakter nyolcadik sorát reprezentáló byte 0-ás bitjébe is 0-át írunk, így a program nem fejezi be a karakter megjelenítését, hanem további sorokat rajzol mindaddig, amíg a karakter végét jelző bitben 1-et nem talál. Ezzel a megoldással élve azonban a speciális karakter kódját követő kódot – kódokat – elveszítjük, hiszen az azok megjelenítéséhez szükséges pontmintázatot tároló 8 byte-os mezőt – mezőket – a speciális karakter pontmintázata részének tekinti a képernyőkezelő program.

A karaktergenerátor szervezése lehetővé teszi, hogy két egymás mellé írt jel összefüggő alakzatot alkosson. Ez annak köszönhető, hogy míg a pontmintázat 6 pont széles, maga a karakter tipikusan csak a bal szélső 5 pontot foglalja el. Ha ebben a 6. oszlopban is van a karakternek értékes pontja, nem lévén már elválasztó mező, a szomszédos karakterek egybefolynak. Az előzőekben leírtak figyelembevételével vizsgáljuk meg pl. az Á betű képét megadó 8 byte-os karaktergenerátor-mező felépítését és az egyes byte-ok decimális értékét (2. ábra). A karaktergenerátor-mező felépítésének megismerése után vizsgáljuk meg, hogyan lehetséges az újonnan definiált karakterek alkalmazása. Ezt végső soron az teszi lehetővé, hogy a képernyőkezelő program a 128–255 decimális kódú karakterek pontmintázatát meghatározó karaktergenerátor-mező kezdőcímét a RAM tárolóterületén őrzi. Így e mező kezdőcíme szabadon változtatható. Ha az ismertetett szabályok figyelembevételével pl. a BASIC programterületen létrehozunk egy, az új karakterek pontmintázatát tartalmazó karaktergenerátor területet és a RAM megfelelő címére – 16459–16460 – elhelyezzük ezen terület kezdőcímét, akkor a képernyőkezelő program a 128–255 kódú karakterek képét erről a helyről fogja megjelenítéskor kiolvasni. Most, hogy megismertük a karaktergenerálást, alkalmazzuk tudásunkat a gyakorlatban! Definiáljunk egy karaktert, amely egy lovat, s rajta egy lovast ábrázol! A pontmintázat tervezésekor hamar beláthatjuk, hogy kielégítő ábrát csak 6 pontnál szélesebb, 8 pontnál magasabb alakzat esetén kaphatunk. A korábbiakban már láthattuk, hogy a Primo felhasználója ezt a problémát könnyen megszüntetheti. Nem egy karaktert kell tehát definiálni, hanem kettőt, melyek közül az egyik az ábra bal, míg a másik a jobb felét ábrázolja, és mind-

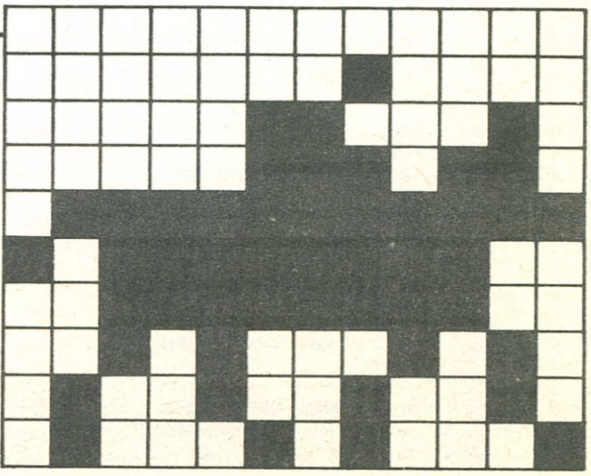


1. ábra

2. ábra

bit	7	6	5	4	3	2	1	0	decimális érték
	0	0	0	1	0	0	0	0	16
	0	0	1	1	1	0	0	0	56
	0	1	0	1	0	1	0	0	84
	0	1	0	0	0	1	0	0	68
	0	1	1	1	1	1	0	0	124
	0	1	0	0	0	1	0	0	68
	0	1	0	0	0	1	0	0	68
	0	1	0	0	0	1	0	1	69

3. ábra



```
100 'Lóverseny  Tiszai Tamás '84.1
110 C=0 : V%=0
120 DIM K%(12) : C=VARPTR(K%(0)) 'Tárcím
130 '128-129 es 130-131 kód definiálása
140 POKE C,0,0,2,2,62,94,30,20,36,35,0,0,0,0,0,0
145 POKE C+16,0,32,68,108,126,120,120,20,36,35
150 POKE 16459,C-256*INT(C/256),INT(C/256)
160 DIM LO%(5) 'Lovak poziciója
170 FOR C=0 TO 5 : LO%(C)=3 : NEXT C
180 PRINT CHR$(12)CHR$(2)CHR$(4)" L ó v e r s e n y  "CHR$(18)CHR$(20)
190 PRINT $4,41,"C" : PRINT $7,41,"E" : PRINT $10,41,"L"
200 FOR C=35 TO 160 : SET (248,C) : NEXT C
210 FOR C=1 TO 5 : PRINT $2*C+1,0,C;CHR$(128)CHR$(130) : NEXT C
220 POKE 16443,PEEK(16443)AND127 'NMI tilos
230 RANDOM : PRINT CHR$(6) 'Előtörlés
240 FOR C=0 TO 500 : NEXT C
250 GOSUB 360 : GOSUB 360 : BEEP 50,400
260 'Lovak mozgatása
270 V%=RND(5) : IF V%=0 THEN 270
280 PRINT $2*V%+1,LO%(V%),CHR$(5)" "CHR$(21)CHR$(128)CHR$(130)
290 BEEP 50+10*V%,3 'Mozgás hangja
300 LO%(V%)=LO%(V%)+1%
310 IF LO%(V%)<>39 THEN 270 'Tovább!
320 PRINT $14,8,"Győzött a"V%". pályán futó ló!"
330 FOR C=0 TO 2000 : NEXT C : PRINT CHR$(22);
340 POKE 16443,PEEK(16443)+128 : OUT 0,PEEK(16443) 'NMI mehet!
350 GOTO 170 'Következő futam inditása
360 BEEP 50,50 : FOR C=0 TO 100 : NEXT C : RETURN
370 END 'Lóverseny
```

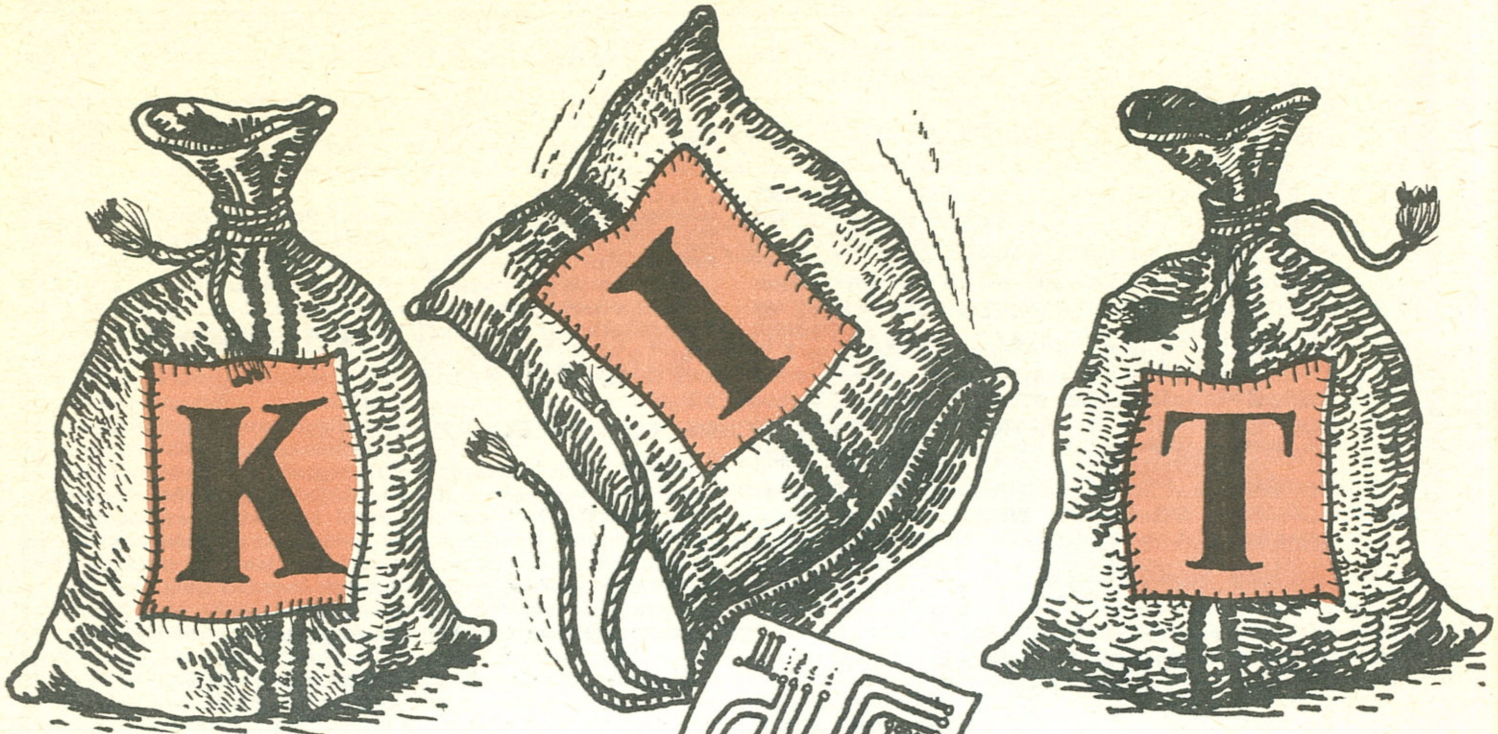
két karakternek a szokásos 8 pontnál magasabbnak kell lennie. A kitűzött célt megvalósító egy lehetséges alakzat pl. 12x10 pontból álló, a 3. ábrán látható megoldás lehet. Az ábrának megfelelő karaktergenerátor-mező decimális kódjai a következők:

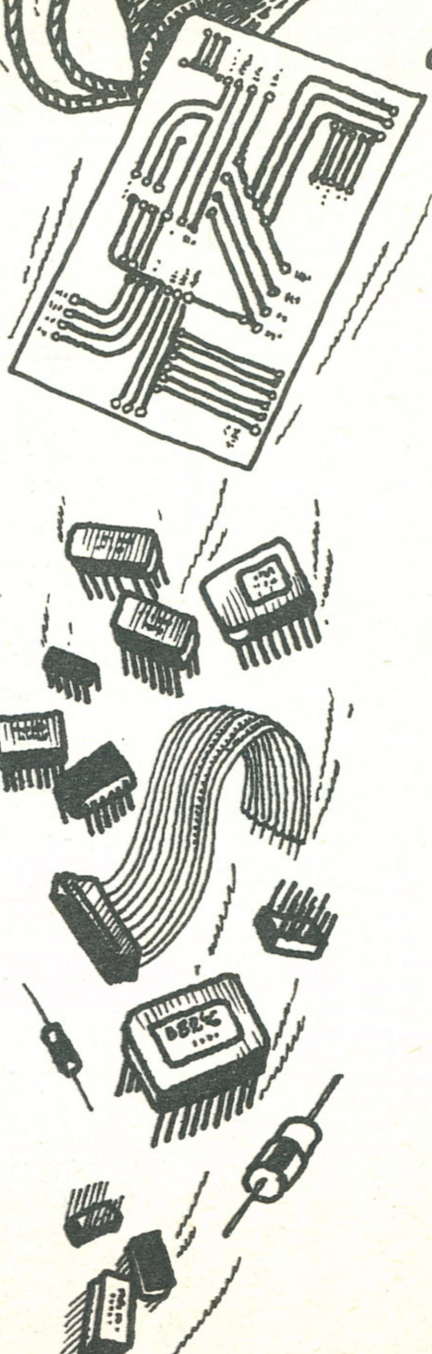
bal oldal	jobb oldal
0, 0, 2, 2, 62, 30, 20, 36, 35	0, 32, 68, 108, 126, 120, 120, 20, 36, 35

Az ábra bal oldalát jelképező karakter kódjának válasszuk a 128-as kódot. Mivel a definiált ábra 8 pontnál magasabb, ez azt eredményezi, hogy a definícióval a 129-es kódot is elfoglaltuk, így az ábra jobb oldalát a 130-as kód képviselheti, amely természetesen a 131-es kódot foglalja még el. Ezek után a teljes ábrát leíró karaktergenerátormező a következő: 0, 0, 2, 2, 62, 94, 30, 20, 36, 35, 0, 0, 0, 0, 0, 0, 0, 32, 68, 108, 126, 120, 120, 20, 36, 35.
(A 11–16. pozícióban álló 6 db 0 a 129-es kódú karakterterület fel nem használt 6 byte-nyi része. E byte-ok megadása azért szükséges, hogy a képernyőkezelő program a 130-as kódú karakter elejét megtalálhassa, vagyis minden karakterkódhoz 8 byte-nyi területet kell definiálni még akkor is, ha azt ténylegesen nem is használjuk ki. Ez alól csak az utolsóként definiált karakter kivétel, amelynek nem használt sorait jelképező byte-ok definiálása természetesen nem kötelező.)
Az elkészített ábra programbeli definiálását és alkalmazását szemlélteti az alábbi egyszerű bemutató program:
A program 120-as sorában álló DIM utasítással lefoglalt K% tárolja majd a definiált karakterek pontmintázatát. A C skalár változó a karaktergenetátor-mező kezdőcímét tárolja.
Ha az új karakterek pontmintázatát tároló karaktergenerátormezőt egy BASIC tömbváltozóban – jelen esetben a K% vektorban – helyezzük el, akkor a program helyes működésének érdekében biztosítani kell, hogy a tömb kezdőcímének meghatározását végző VARPTR függvény alkalmazása után a programban korábban még nem használt skalár változó már ne szerepeljen. Egy ilyen változó megjelenése ugyanis azt eredményezi, hogy megváltozik a tömbváltozók tárolón belüli kezdőcíme, tehát a képernyőkezelő program nem képes a karaktergenerátormezőt megtalálni.
Ezt a problémát úgy szüntethetjük meg, hogy a program elején – mint a 110-es sorban is látható – egy fiktív értékadásban felsoroljuk a programban alkalmazott valamennyi skalár változót. A karaktergenerátor-mező feltöltése a 140-es sorban történik meg. A 150-es sorban levő POKE utasítás állítja be a 128–255 kódú karakterek karaktergenerátor mezejének kezdőcímét tartalmazó rendszerváltozót a megfelelő értékre. (Ezzel egyidejűleg a Primo felhasználói kézikönyve 3. mellékletében 128–151 kódok alatt szereplő karaktereket a gép „elfelejti" mindaddig, amíg a RESET gombot meg nem nyomjuk. Ekkor viszont az általunk definiált karakterek szűnnek meg a program újabb futtatásáig.) A program 220-as és 340-es sorában álló utasítások hatásának ismertetésére egy későbbi cikkben térünk vissza. (Az említett sorok a programból akár el is hagyhatók.) Ha viszont szerepelnek a programban, akkor a RESET gomb időszakosan hatástalanná válik. Ezért – most nem részletezett okokból – az egyébként végtelen ciklusban futó programot a BRK gombbal csak akkor szakítsuk meg, amikor éppen befejeződött egy futam. (Ellenkező esetben a RESET gomb aktivizálása nem történik meg.)
A mintaprogram megfelelő részeinek megértése után bármely Primo-felhasználó saját programjában is eredményesen alkalmazhat újonnan definiált karaktereket, melyek érdekesebbé, gazdagabbá tehetik elkészített programjaikat.

Tiszai Tamás

Amikor elraktározol valamit
a gép memóriájában,
raktározd el saját memóriádban,
hogy hova tetted!
(Leo Beiser első számítógép-axiómája)





KIT – angol szó – jelentése: cókmók, készlet, felszerelés. Mikroszámítógépes körökben a szó közismert, bár valószínűleg az eredeti jelentés ismerete nélkül. Azokat a csomagokat nevezik így, amelyek egy-egy számítógép megépítéséhez szükséges anyagokat, alkatrészeket tartalmazzák. A szónak már csak azért sincs magyar megfelelője, mert mind ez idáig ilyen KIT-ben építhető számítógép nem volt hazánkban. Magyar megfelelőt ugyan mi sem találtunk a szónak – nem is nagyon kerestünk –, viszont „megtaláltuk" az első, majdnem KIT-ben építhető magyar számítógépet. Konstruktőre: Lukács József és Lukács Endre.

VALLOMÁS

Amikor az első számítógépemet konstruáltam, még élénken élt bennem egy, azóta már nem létező amerikai cég, az OHIO SCIENTIFIC példája. Hogy mi lett velük? Tönkrementek, más nagy cégek felvásárolták. Hogy miért? Ez volt az a vállalkozás, amelyik 1978-ban 300 dollárért kínált egészen kitűnő számítógépet CHALLANGER néven (akkor a többiek, az APPLE, TANDY, COMMODORE még mind 700 dollár fölött voltak!)

Én elfogadtam a CHALLANGER kihívását, és elhatároztam, hogy a magyar ifjúságnak is megpróbálom megteremteni az elérhető számítógép lehetőségét. Magunkat – mert mindehhez lelkes és fáradhatatlan kollégára az öcsém személyében találtam, elneveztük HOMELAB-nak, és munkához láttunk valóban Házi Laboratóriumunkban.

Ha az első gép még nem is, de az 1982 tavaszán elkészült második típus már megfelelt azoknak a követelményeknek, melyet magunk elé állítottunk. A történet innentől kezdődően ismert. Az eredmény kb. 200 gép. A fele a HCC klubban megépített ***HOMELAB II.*** a másik fele pedig az AIRCOMP 16 névre keresztelt BOSCOOP-PERSONAL gyártmány. A gép forgalmi ára 27 000 forint lett, amiről meg kellett állapítanom: túl sok ahhoz, hogy egy magamfajta diák a magnóra félretett pénzéből megvehesse.

A HOMELAB III ezért készült el 1983 nyárra. Ez a második gép mindenféle tapasztalatait összegezve, az akkori alkatrészkínálatot is figyelembe véve 10 ezer forint alatti alapgépet jelentett. A tapasztalatokhoz az is hozzátartozott, hogy sajnos állami gyártó cég még a legjobb indulat mellett sem tudja majd teljesíteni az általam kitűzött árat, így elsősorban saját gyártásra gondoltam. A korábbi gép hasznát alkatrészbe, dobozba, billentyűzetbe fektetve vártam, hogy kirukkolhassak a masinával. Ehelyett az történt, hogy a pénzem elfogyott, és hiába volt 100 gépre való RAM-om, NAND kaput nem raktároztam, márpedig a végén már a legelemibb alkatrészeket sem lehetett beszerezni; arról nem is beszélve, hogy ami akkor még filléres Tungsram-gyártmány volt, az most nyugati, és ötször annyiba kerül. Keserűen kellett rádöbbennem, hogy ebből így nem lesz semmi, így én is az OHIO SCIENTIFIC sorsára jutok, bár engem itt egyáltalán nem a kíméletlen konkurenciaharc tesz tönkre.

Elhatároztuk tehát, hogy nem takargatjuk tovább immáron bölcsődés korúvá cseperedett gyermekünket. Ha nem lehet belőle gép, legyen belőle KIT. Méghozzá magyar KIT. Olyan, amihez a vevő kénytelen beszerezni az alkatrészeket. Jól működő hivatalos csatorna híján turistaút, kiküldetés, esetleg segítőkész rokon jöhet számításba a hazai kiskereskedelem mellett. Ennek persze meglesz az az előnye, hogy senki nem tudja majd egyszerre megterhelni a pénztárcáját, tehát ez tényleg zsebpénzből építhető-építgethető gép lesz. Én csak arra vállalkozom, hogy az építéshez szükséges leírást, nyomtatott áramköri lemezt (NYÁK), a dobozt és a billentyűzetet biztosítom, és EPROM-ba égetem a BASIC interpretert. Ez talán nem túl sok, de a hiányzó rész egy gép esetében lelkesedéssel kipótolható. Tulajdonképpen így született meg a HCC-ben a korábbi 100 gép is, és a klub részt vállal ebből az akcióból is. A HOMELAB szekció vállalta, hogy amenynyire szűkös lehetőségeiből telik, bábáskodik a HOMELAB III fölött is. EPROM égetéssel, élesztési tapasztalatokkal, információkkal segítik majd az építkezők munkáját, és a készülő szoftvereknek is fórumot biztosítanak.

HOMELAB

ÉS MIT TUD A GÉP?

Úgy terveztük, hogy lehetőleg mindent tudjon vagy tudhasson a megfelelő bővítésekkel. Maga a nyák olyan, hogy statikus és dinamikus memória egyaránt kerülhet bele, így a gép 2 K-tól 64 K-ig tartalmazhat RAM-ot. Lehetséges, hogy valaki 2 K-val indul, és ahogy nőnek az igényei (és lehetőségei) bővíti a gépét, pusztán újabb IC-k beépítésével. A címkiosztás tetszőlegesen programozható, így RAM-nak is és a rendszerprogramot tároló EPROM-nak is többféle alkatrész használható (2716, 2732, 2764, 6116, 4116, 4164 stb.). CMOS-RAM (pl. 6116, 5516) használata esetén „nem felejtő" RAM-ot is lehet csinálni. A NYÁK-on elhelyezhető 2 gombakku kb. fél évig kitart. Most lássuk a displayt. Egy kép 32 sorból áll, és soronként 32 vagy 64 betű lehet. Ezt néhány átkötéssel fixen lehet beállítani vagy egy kapcsolóval átkapcsolhatóvá lehet tenni. Mi a 64 betűs változatot ajánljuk, de ez jó minőségben csak videóbemenetről működik. A legtöbb tranzisztoros tv-nél egész egyszerű kihozni egy videóbemenetet, de gondolva azokra, akik nem akarnak kotorászni tv-jükben, van egy modulátor is a NYÁK-on. Sajnos beépített teljes grafika nincs, de cserébe a karakterkészletet úgy próbáltuk megválasztani, hogy a magyar betűkön túl tartalmazzon egy kvázigrafikus karakterkészletet és egyéb grafikus jeleket is, amivel koordinátarendszert, oszlopdiagramot, digitális idődiagramot, nyulat, kutyát, vitorlást stb. lehet rajzolni. A billentyűzet hardverje 8x10-es mátrixot bír el. Mi ebből csak 60 gombot használtunk fel. Ebben benne vannak a magyar betűk, független cursormozgatások, és két funkciós gomb is.

Így a SHIFT megmarad a nagybetű/kisbetű váltásnak, míg a funkciós gombbal meghatározott jelek vagy szövegek hívhatók elő az egyes billentyűkkel.

Kazettás magnó természetesen kapcsolható a géphez, és a beolvasó elektronikája nagy- és kisjelű magnókhoz egyaránt alkalmas.

Nem esett még szó róla, pedig fontos: van a gépben egy PIO is. Ez a felhasználó által tetszőlegesen programozható, így a nyomtató lekezelésén túl bármilyen 16 bites mérésre, vezérlésre stb. felhasználható, vagy játékbemenetnek alkalmas.

A gép áramellátását egy 5 V max. 800 mA stabilizált tápegység biztosítja. Ezt egyelőre mindenkinek magának kell megoldani.

A hardverhez tartozik az is, hogy a memóriafelosztást adó PROM-címgenerátor egyszerre kétféle kiosztásra is beégethető, és ez programból átkapcsolható. Ez biztosítja, hogy 64 K esetén a ROM helyén eltűnő memória a másik lapon elérhető, ill. ezzel lehet megoldani CP/M fogadásának előfeltételét, a csupa RAM felosztást. A CP/M már működik a gépen, de nem elégedtünk meg ezzel. Most úgy próbáljuk átalakítani, hogy további 64 K RAM-mal a rendszer lényegesen felgyorsuljon. (Hogy floppy honnan lesz hozzá azt nem tudom!)

Van a HOMELAB III-hoz egy beszélő bővítés is. Ezt a géphez csatlakoztatva a gép tetszőleges BASIC-ben beprogramozott vagy beírt magyar szöveget ki tud mondani, sőt még hangsúlyozni is képes a mondatokat. Alkatrészigénye minimális, egy chip kivételével minden megkapható hozzá itthon is.

Színes grafikai bővítést nem szándékozunk építeni hozzá, ellenben a klubtagok már látták működni az 512x256-os fekete-fehér grafikát. Előbb-utóbb ebből is KIT lesz, de erre még várni kell.

A SZOFTVER

A gép alapszoftverje tulajdonképpen megegyezik a korábbi gép BASIC-jével. Persze eltérések akadnak, hiszen mások a gép lehetőségei, de a kompatibilitás (felülről) biztosított.

A HOMELAB II (AIRCOMP 16) programjai kazettáról gond nélkül beolvashatók és futtathatók. Emiatt a kompatibilitás miatt itt most nem is részletezem az alap BASIC tulajdonságait. Viszont ismertetném az új utasításokat és lehetőségeket. Bevezettük az INKEY és INKEY$ változókat, és a REM utasítást. Ezeket eddig, csak kicsit körülményesen lehetett pótolni. A BEEP új utasítás elfütyüli a mögéje írt stringet. Egy-egy karakter, egy-egy hangnak felel meg, és 32-féle ritmus is beállítható, akár menet közben is.

Bevezettük a CUR X, Y-t is ami a PRINT-ben és INPUT-ban a cursort a képernyő tetszőleges pontjára állítja. A PRINT és a LIST printer is tud kezelni.

A MON új utasítás MONITOR-parancsokat hajt végre a BASIC-ben. A MON után írt stringet monitorparancsként végrehajtja, és visszatér a BASIC-be. A MONITOR egyébként a „hagyományos" HOMELAB MONITOR kiegészítve a nyomtatót kezelő rutinokkal. Ezeken kívül két további utasítást láttunk el operandussal. Az egyik a RESTORE A, amelyik az A-adik sorban elhelyezett DATA utasításra állítja vissza a READ olvasómutatóját. A másik a NEW utasítás, ahol meg lehet adni azt a decimális memóriacímet, ahonnan a BASIC-programot felépíti a rendszer. A PLOT utasítás viszont szűkült. Mivel csak kvázigrafika van, 128x96-os felbontás érhető el vele. A BASIC-en kívül már elkészült a Bővített Monitor és Assembler, a dupla pontosságú aritmetika, és a Bővített BASIC.

A BASIC bővítés hexadecimálisan és binárisan is számol, van benne kerekítő függvény, MIN, MAX, FRA függvény, és a számok formattált kiíratása is lehetséges. Egyszerre több független program is lehet a gépben, lehet programokat összeilleszteni, és VERIFY parancs is van.

Az ASSEMBLER egy egyszerű, de nagyon jól használható, sorról sorra és oda-vissza fordító assembler, a duplapontos pedig 16 jegyre képes számolni.

Láttunk már a gépen működni FORTH-ot és szimbolikus Assemblert is, és hamarosan ezek is standard programbővítések lesznek. Megjelent már az első program füzet is, ami a klubtagok, és az országban elszórt lelkes Homelabosok programjait tartalmazza.

MI KELL HOZZÁ?

Hát az attól függ. Mint a hardver leírásból látható, ez nem egy merev gép. Itt az alkatrészigény a kitűzött céltól függ.

Ezek szerint a 2, 16 és 64 K-s kiépítések alkatrészköltsége durván 3000, 4500 és 9000 Ft. Hangsúlyozni kell azonban, hogy

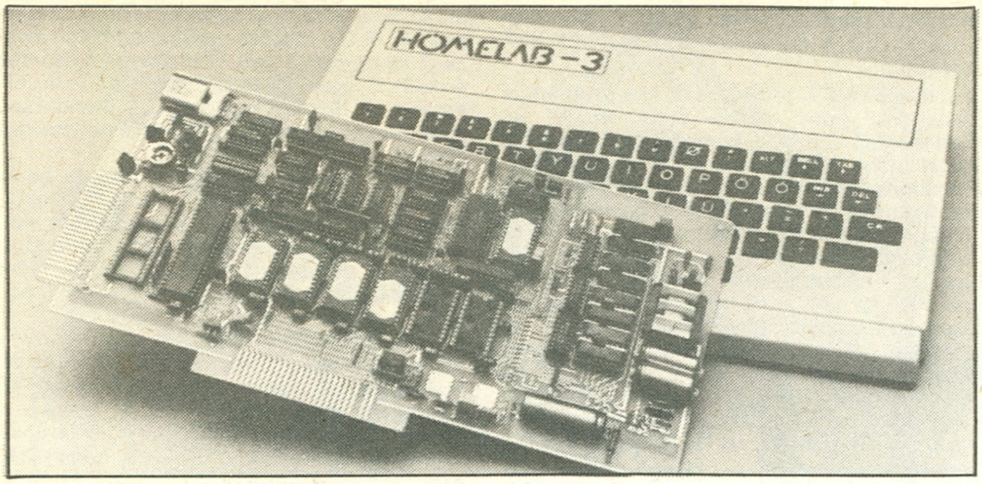
A HOMELAB 3 kulcsszó-készlete

ABS()	AND	ASC()	ATN()
BEEP	CALL	CHR$()	CONT
COS()	CR	CUR	DATA
DIM	END	EXP()	EXT
FOR	FRE()	GOSUB	GOTO
HM	IF	INKEY	INKEY$
INPUT	INT()	LEN()	LFT$()
LIST	LOAD	LOG()	MID$()
MON	NEXT	NEW	NOT
ON	OR	PEEK()	PI
PLOT	POINT()	POKE	POP
PRINT	READ	REM	RESTORE
RETURN	RGH$()	RND()	RUN
SAVE	SGN()	SIN()	SQR()
STEP	STR$()	TAN()	THEN
TO	USR()	VAL()	

A HOMELAB 3 magyar szókészlete

ABSZ()	ÉS	KOD()	ATG()
HANG	REND	JEL$()	TOVÁBB
COS()	SZIN	HELY	ADAT
TÖMB	VÉGE	EXP()	BÖV
CIKLUS	TÁR()	ALPROG	FOLYT
LEG	HA	GOMB	GOMB$
KÉRDEZ	EGÉSZ()	HOSSZ()	BAL$()
LISTA	TÖLT	LN()	KÖZ$()
MONITOR	KÖV	ALAP	NEM
ELÁG	VAGY	VESZ()	PI
RAJZ	PONT()	TESZ	KILÉP
IR	OLVAS	MEGJ	ÁLLIT
VISSZA	JOBB$()	VÉL()	RAJT
TÁROL	SIGN()	SIN()	GYÖK()
LÉPÉS	SZÖV$()	TG()	AKKOR
AMIG	GÉPI()	ÉRTÉK()	

SZUPER HOMELAB



terheltek. Nem esett szó eddig a billentyűzetről. Itt sajnos valódi jó megoldás híján több változat is lehetséges.

1. Sikerül beszerezni 60 db nyomógombot. Ekkor a megfelelő mátrixot összehuzalozva 100%-os megoldáshoz jutunk. Vigyázat! A mechnikus nyomógomb itt jobb, mint a Hall-generátoros. Ez utóbbinál a mátrix megvalósításához további hardver is szükséges.
2. Fóliatasztatúra, melynek a NYÁK-lemeze és öntapadós borítása rendelkezésre áll, viszont ezek összeállítását az ezzel foglalkozó cégek túl drágán vállalják. Így marad a kísérletező kedv a fent említett anyagokból.
3. Gumimembrános-vezetőgumis tasztatúra. ezek az árak legalább ±50%-os hibával

Ez egy saját fejlesztésű billentyűtípus, és a fröccsöntött dobozzal együtt a SPECTRUM-hoz hasonló megoldást ad.

ÉS MIBE FOG KERÜLNI?

A KIT-et is több változatban kínáljuk. Az itt felsorolt tételek külön-külön is igényelhetők.

1. alap-NYÁK+dokumentáció+alapszoftver beégetés: ***1500 Ft.***
2. Fóliatasztatúra NYÁK és öntapadós borító: ***800 Ft.***
3. gumimembrános billentyű+doboz (ez csak együtt működik): ***1800 Ft.***
4. Assembler program (2K) beégetés ***400 Ft.***
5. Bővített Basic (2K) beégetés ***400 Ft.***
6. Duplapontos aritmetika (4K) beégetés ***400 Ft.***
7. CP/M illesztőkártya (+ rendszerlemez) ***3000 Ft.***
8. Beszélő kártya + programbeépítés (8 K) ***3000 Ft.***

Mindezeket a részegységeket Kardos József kiskereskedő boltjában megkaphatják.

3. Érdeklődni és a klub munkájába bekapcsolódni a kedd esti összejöveteleken lehet.: Belvárosi Művelődési és Ifjúsági Ház HCC HOMELAB Klub

Budapest V., Molnár u. 9.

Összejövetelek minden kedden 18–21 óráig.

BIZTATÓ

Az eddig elkészült kb. 300 gép éledése viszonylag simán ment, kizárólag az összeszerelési hibák kijavítására korlátozódott. Pontos, körültekintő építés esetén beállítandó, behangolandó alkatrészek nincsenek, minden biztonságosan működik.

Végezetül hadd fejezzem be még egy személyes gondolattal. Szeretném ha egy-két év múlva egy középiskolás úgy kezdené a cikkét: „Jó volt az a HOMLEAB III. de én már tudok egy jobbat!"

dr. Székely Jenő



SZAKKÖRÖKNEK!

BEVEZETŐ

GÉPI KÓDÚ UTASÍTÁSOK

KÓDTÁBLÁZAT

A személyi számítógépek megjelenésétől kezdve egyre többen jutnak gép közelébe, tanulnak, játszanak. Sokan megismerkednek a BASIC nyelvvel, programokat írnak. Az elkészült programok gyakran más hatást keltenek, mint amit a tervezőjük megálmodott, pl. szembántóan villódzva változnak az ábrák, vagy csigalassúsággal mozognak a versenyautók. Ilyenkor nem mindig nyilvánvaló, hogy programozási ügyetlenség, a számítógép fogyatékossága vagy a BASIC korlátjai okozták a kellemetlen meglepetést. Saját magunk és kedvenc gépünk tudásába vetett hit hamar kimondatja: ***gépi kódban kell megírni a programokat,*** *de legalábbis egy részét. Ezt a véleményt az is erősíti, hogy a közkedvelt látványos játékprogramok többsége gépi kódban készült.*

Nemegyszer hallottam már, hogy a gépi kód „magasabb szintű", mint a BASIC. Ez persze nézőpont kérdése. Lehet, hogy valaki egy zsák alkatrészt magasabb szintű kiszolgálásnak tekint, mint egy kész autót, mondván: többféle összeépítési stb. lehetőség van benne. De a számítástechnikai szaknyelv a BASIC nyelvet mondja magasabb szintűnek, mert használatával rengeteg rabszolgamunkától, gépies tevékenységtől mentesülünk, nagyobb hatáskörű, átfogóbb utasításokat adhatunk, mint gépi kódban. A technikai részletkérdések a BASIC-programozó előtt rejtve maradnak. S hozzá kell még tennem: a BASIC nyelv éppen csak hogy rászolgál a „magasabb szintű programnyelv" megtisztelő címre.

Éppen egyszerűsége miatt sikerült aránylag kis méretű memóriákba belepréselni azt a tudást, amire a gépnek szüksége van a BASIC nyelv megértéséhez, de se szeri, se száma a legalább ilyen fejlett programnyelveknek. Valószínűleg többen kaptak már más programozási nyelvhez kazettát kisszámítógépükhöz. Semmi technikai akadálya nincs annak, hogy ezeket a BASIC-hez hasonlóan a gépbe beépítve árulják.

A gépi kódú programozással kapcsolatban tisztában kell lennünk azzal, hogy több nehézséggel jár, mint a BASIC használata. Igaz ugyan, hogy egy jó assembler-editor birtokában megszabadulhatunk a gépi kódú programozással járó kényelmetlenségek jelentős részétől – de azért ránk is jut még bőven. Mindenekelőtt sok olyan adatot kell tudnunk gépünkről, amelyek a BASIC-használók számára lényegtelenek. Esetleg a gép áramköreiről (IC-iről), kivitelezéséről is pontosabb ismeretekre lesz szükségünk.

Milyen előnyöket várhatunk a gépi kódú programozástól?

Elsősorban azt, hogy a gépünk minden beépített hardver lehetőségét kihasználhatjuk. Ez általában a gép és a külvilág közötti kapcsolattartás vonatkozásában jelentős. Egyes programjaink valóban gyorsabbak lesznek – de gépi kódban is könnyű ügyetlen, lassú programot írni. Gépi kódban megtehetjük, hogy egyes résztevékenységek megoldására számunkra kedvezőbbnek tűnő utat választunk, mint a BASIC interpreter tervezői tették.

De semmiképp se feledjük: ***a gépi kód nem csodaszer!***

ZX81 gépünk gépi kóddal ingerelve sem fog 16 színű finomgrafikás képernyőt adni. Ne fűzzünk hozzá csalfa reményeket – de akinek van kedve, ideje, jöjjön! Lássuk a medvét!

Leírásunkban a Z80 mikroprocesszorról lesz szó.

Nem, nem nyomdahiba! Valóban Z80, és nem ZX80 vagy 81 – azok személyi számítógépek – igaz, mindkettő Z80-nal működik, az a lelke. De Z80-nal dolgozik a ZX Spectrum és a HT 1080Z is. Ez azt jelenti, hogy ezeknek a gépeknek közös a gépi kódú nyelve. Ugyanezt a nyelvet „beszéli" az AIRCOMP, a Boscoop gépe is, meg a nemrég megjelent PRIMO. Ez azt jelentené, hogy ha már a BASIC programok nem cserebérélhetők e gépek között, legalább a gépi kódú programok átvehetők? Sajnos, a gépek is úgy vannak ezzel, mint az emberek: lehet, hogy egy nyelvet beszélnek, de attól még előfordul, hogy mindenki mást akar mondani – vagy ami még rosszabb, ugyanaz a szó vagy mondat egy más környezetben mást eredményez, mást jelent. A továbbiakból ennek is kiderül az oka.

Maga a Z80 mikroprocesszor egy elég nagy tudású, sok műveletet ismerő „számítógép". Még saját belső memóriája is van, igaz csak 208 bit (nem byte!), amit a programozó közvetlenül használhat. Egyik gyártójának leírása szerint 158 műveletet ismer, de birtokomban van egy olyan lista is, amelyen 694 utasítás szerepel – s még ez sem teljes: két sorozatot „nem hoztak nyilvánosságra". A ZX81 és Spectrum kézikönyvekben levő lista 256 sorban tartalmaz 1–3 utasítást. Most melyik kiadványnak van igaza? Az elírásoktól eltekintve, talán mindegyiknek. Bizonyos utasításcsaládot az egyik lista egynek számol, a másik pedig tételesen felsorolja. De ez egyben jó hír a gépi kódot tanulóknak: ügyes csoportosításban egyszerre nagy „családot" lehet megjegyezni.

Kívülről nézve a Z80 egyetlen, nem is olyan nagy méretű IC: kb. 5,3 cm hosszú, 1,5 cm széles, 40 lába van. Belsejében egy kis szilíciumlapocskán néhány tízezer tranzisztornak megfelelő kapcsolás található. Az ember azt hinné, hogy ez a sok kapcsolási elem épp hogy elfér a tokban – de nem így van! Az IC tulajdonképpen más okokból ilyen nagy kívülről (pl. hőleadás, meg a lábak is szabványfoglalatba csatlakoznak stb.): az a bizonyos „morzsa" fél négyzetcentiméter sincs!

Miért van 40 láb? Ezek közül kettő a tápfeszültség (5 V), egy az úgynevezett órajel (néhány millió „négyszög jel" másodpercenként). 16 lábon küld ki olyan jeleket, amiből a hozzá kapcsolódó, pl. a memória-áramkörök megtudják, melyik rekeszükkel akar dolgozni a Z80, 8 lábon zajlik a rendszeres adatcserebere. A többi kivezetéseken fontos vezérlőjeleket ad/vesz a mikroprocesszor. Például meg kell mondja: memória vagy input/output egység az, amivel foglalkozni akar, írni vagy olvasni szándékozik stb.

A Z80 mikroprocesszor másodpercenként néhány milliószor megvizsgálja a lábain levő feszültségértéket (ezek jelentik számára a külvilág ingereit), s a következő pillanatokban ezektől függően végzi tevékenységét. Ennek eredményeként egyes kivezetésein „magas" (több mint 2,4 V), vagy „alacsony" (max. 0,4 V) feszültség jelenik meg.

Amikor bekapcsoljuk a gépünket, vagy amikor megnyomjuk a RESET gombot, a Z80 egy erre szolgáló lábán érzékeli: most törölnie kell regisztereit (kis belső memóriarekeszeit), többek között az úgynevezett „programlépés-számlálót" (PC: Program Counter). Ez számunkra azért érdekes, mert a Z80 minden utasítás végrehajtása után a következő tevékenységbe kezd:

A PC tartalmát a 16 címvezetéken a kapcsolódó áramkörök tudomására hozza, megfelelő vezetékein bejelenti, hogy memóriából akar olvasni, majd egy kicsit vár. Ezalatt a memória-áramkör (IC) felébredhet, tudomásul veszi, hogy hozzáfordultak, érzékeli, hogy író vagy olvasó műveletet kell végeznie, s azt is, hogy melyik rekeszéről van szó – ezt mondja meg a 16 címvezeték. Induláskor a Z80 állapota azt jelenti, hogy a 0 című memóriarekesz tartalmát kérdezi. Ennek megfelelően a memória áramkör a 8 adatvezetékre írja a 0 című rekeszének tartalmát. Most ismét a Z80-on a sor: ezt a jelet mint az elvégzendő utasítás kódját értelmezi, s hozzákezd annak végrehajtásához. A továbbiak természetesen attól függnek, hogy mi volt a beolvasott kód. Akit érdekel, hogy gépén mi is ez az első kód, kérdezze meg gépét:

```
PRINT PEEK (0)
```

Természetesen most nem alacsony és magas feszültségértékek jelennek meg a képernyőn, hanem egy 0–255 közötti egész szám. De ez már a BASIC tulajdonsága – pontosabban a PEEK függvénye – amely a memóriarekeszek tartalmát mindig úgy jelzi ki, mintha az egész számot jelentene, a 0–255 számkörből. Arról persze szó sincs, hogy a Z80 ebbe belekeveredne: ő most utasítást vár, akként is értelmezi a beolvasott jelet. Persze a BASIC-ben is lehetne egy olyan függvény, amely nem szám, hanem utasítás-értelmezésben fejti vissza a beolvasott jelet, sőt azt ki is írhatná ékes magyar nyelven, de ez nem szokás. Nekünk persze ez most hiányzik. Mások is jártak már így – meg is írtak már nem egy olyan programot, amely ha nem is magyarul, sőt nem is angolul, de legalább angol rövidítésekkel kiírja, hogy mit jelent az egyes memóriarekeszekben levő jel, ha utasításkódnak tekinti a processzor. Az ilyen programokat nevezik DISASSEMBLER-nek. Egy ilyen program tehát gépi kódról az angol rövidítéseket használó ASSEMBLER-re fordít. Ez már közelebb áll az emberhez, meg lehet szokni. Egy disassembler birtokában sokat tanulhatunk a profi programozóktól, elleshetjük műhelytitkaikat, megfejthetjük – időnként elég fáradságos munkával – hogyan dolgozik a gépünk, saját gépi kódú próbálkozásainkat egy ASSEMBLER-EDITOR program segíthet a géppel megértetni. Akinek ilyen programok nem állnak rendelkezésére, azok is követhetik sorozatunkat. Elmondjuk majd, hogy ilyen csodaprogramok nélkül, egyszerű eszközökkel – igaz, egy kicsit több munkával – hogyan közölhetjük gépünkkel gépi kódú utasításainkat.

1. ROM ÉS RAM

Már említettük, hogy a Z80 mikroprocesszort a kivezetésre adott „magas", ill. „alacsony" feszültségszintekkel lehet a kívánt művelet elvégzésére ösztökélni. Azt, hogy ezek a vezérlőjelek a megfelelő pillanatban a megfelelő értéket vegyék fel, a Z80-hoz kapcsolt memória-IC-k biztosítják. Ehhez természetesen ezeket a memória-IC-ket megfelelően elő kell készíteni. A BASIC értelmező programjával nincs is semmi gondunk: ezt az IC gyártásakor „beégették" az erre alkalmas ROM-ba. A ROM betűszó: Read Only Memory = csak olvasható memória. Az elnevezés arra utal, hogy ez a memóriatípus olvasható, de pl. programból nem írhatunk bele – egyszerűen nincs felkészítve arra, hogy tartalmát megváltoztathassuk. Előnyös tulajdonsága, hogy kikapcsoláskor sem felejti el a beégetett programot. Elég bekapcsolni a berendezést, és a gép máris „érti" a megfelelő (esetünkben: a BASIC) nyelvet.

Saját gépi kódú programjainkat nem tervezzük ilyen halhatatlannak: megelégszünk azzal, hogy ha begépelés, esetleg kazettáról való betöltés után futtatható. Ezért RAM típusú memóriába írjuk – ugyanúgy, ahogy a BASIC értelmező is RAM-ba írja BASIC programunkat vagy a különböző változókat. A RAM is betűszó: Random Access Memory= véletlen elérésű memória. Az elnevezés egészen más szempontokat

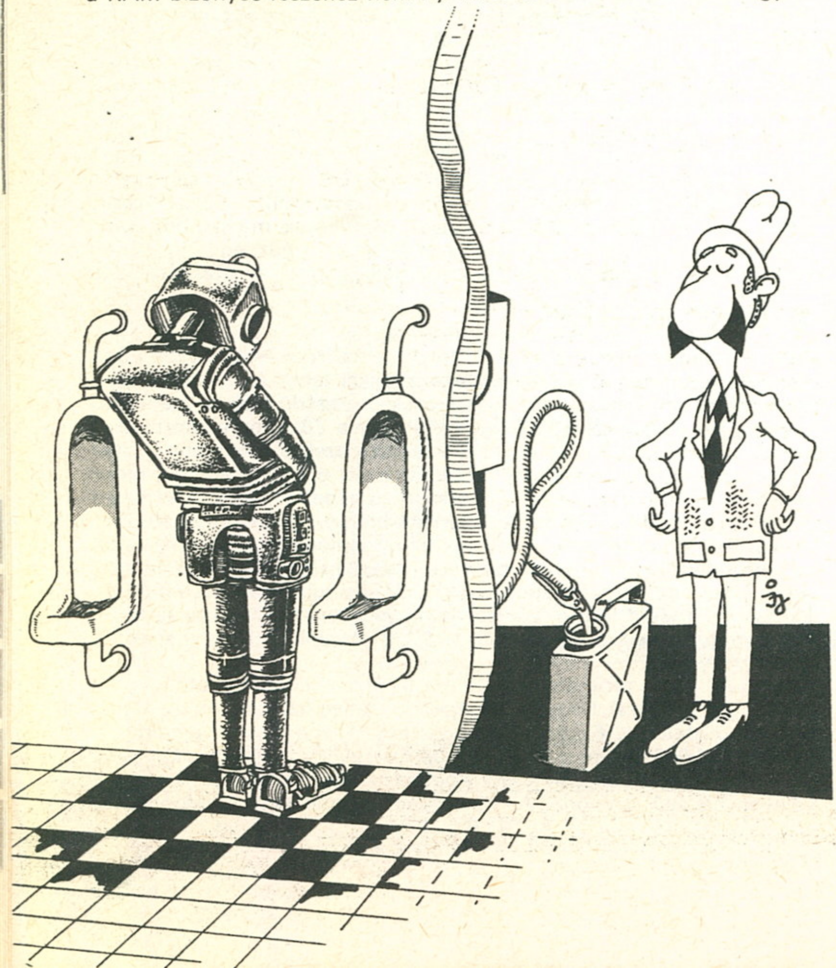
tükröz, mint a ROM neve: azt hangsúlyozza, hogy ha akarom, kiolvashatom belőle pl. a 713. rekesz tartalmát anélkül, hogy az előtte álló 712 memóriarekeszt kiolvasnám. Számunkra is lényeges ez a tulajdonság, de az is, hogy írni lehet bele. Ért olyan utasítást, hogy pl. a 713. rekesz tartalmát írja át 85-re. Ilyenkor a régi tartalom visszavonhatatlanul elvész.
Az eddigiekből legalábbis gyanús lehet: a gépi kódú program írójának tudnia kell, hogy mely memóriarekeszek tartoznak ROM, ill. RAM típusú áramkörökhöz. Ez így is van! Ezért a leggyakoribb géptípusokhoz az 1. táblázatban megadjuk ezeket a címeket. Látható, hogy a címeket kétféleképpen is leírtuk: tízes és tizenhatos számrendszerben. Majd később meglátjuk, melyik mikor használható.

	ROM	RAM
HT 1080Z	0000–35FF	3C00–7FFF
	(0–13824)	(15360–32767)
ZX81 alapgép	0000–1FFF	4000–43FF
	(0–8191)	(16384–17407)
ZX81+16KRAM	0000–1FFF	4000–7FFF
	(0–8191)	(16384–32767)
Spectrum 16K	0000–3FFF	4000–7FFF
	(0–16383)	(16384–32767)
Spectrum 48K	0000–3FFF	4000–FFFF
	(0–16383)	(16384–65535)
Primo 16K	0000–3FFF	4000–7FFF
	(0–16833)	(16384–32767)
AIRCOMP–1	0000–3FFF	4000–7FFF
	(0–16383)	(16384–32767)
AIRCOMP–32	0000–3FFF	4000–C3FF
	(0–16383)	(16384–50175)

Itt említem meg, hogy a K az 1024 jele a számítástechnikában. Így 58 K = 58·1024 = 59 392, 64 K = 64·1024 = 65536. Tekintve, hogy a memóriák címzését 0-tól szokás kezdeni, 64 K RAM a 0–65535 címek használatát teszi lehetővé.

2. Hová írjuk a gépbe a gépi kódú programunkat?

Sajnos nem elég a RAM címeket ismernünk – azt is tudnunk kell, hogy ezek közül melyiket használhatjuk szabadon, és melyiket használja a gép, pontosabban a BASIC. Ebben is eltérnek a gépek. Kialakult azonban néhány olyan módszer, amely megfelelő módosítással csaknem minden gépen alkalmazható, és a gépi kódú rész(ek) a BASIC-kel együtt használhatók.
Talán az a legszimpatikusabb változat, mikor a BASIC-kel közöljük: a RAM bizonyos részéhez nem nyúlhat. Ez a kedvelt módszer egyes



gépeknél kellemetlenné válik, ha a nagy fáradsággal összehozott gépi kódú programot kazettára akarjuk menteni: a BASIC esetleg komolyan veszi, hogy ehhez a területhez semmi köze – nem is hajlandó SAVE-elni.
Egy másik módszer: a gépi kódú programot REM utasításokba írjuk be, általában egyszerű segédprogram segítségével. Ha van kész segédprogramunk, kis mértékben pótolja a már korábban említett ASSEMBLER-EDITOR-t. Ha nincs, gépünk memóriatérképét, kódtáblázatát és programtárolását ismerve megírhatjuk. Balogh Györgyi cikkére (l. az Ötlet 1984. március 1. (88.) számában) és a számtalan hozzászólásra gondolva ez nem lesz reménytelenül nehéz feladat. Mi is közlünk néhány géphez ilyen programot a következőkben.
A REM-ben tárolt gépi kódú részt a programmal együtt gond nélkül lehet kazettára kivinni, ill. onnan beölteni. Esetleg a hibák javítására felhasználhatjuk a BASIC-sorok javítására szolgáló lehetőségeket. Ügyelnünk kell azonban arra, hogy egyes számítógépek szerkesztéskor akaratunktól függetlenül is módosítják a REM sorokat! Ugyancsak problémát okozhat az is, hogy ha a gépi kódú rész pl. a
130 REM . . .
programsorba kerül, akkor minden olyan változtatás, amely kisebb sorszámú BASIC programsorra vonatkozik, a memóriában eltolhatja (előrébb vagy hátrébb helyezheti) a gépi kódú utasításokat. Ez bizonyos esetekben kellemetlen hibák forrása lehet. Ezért az 1 REM, sőt a 0 REM sorba írunk elsősorban. Az sem előnyös, hogy programunk listázásakor automatikusan megjelenik (elég érthetetlen formában) a gépi kód nyoma.
Egy harmadik módszer szerint a gépi kódú programot egy BASIC program alakítja ki egy karakterváltozóban vagy egy tömbben. Ügyelni kell arra, hogy értékadásnál általában nemcsak a szóban forgó változó változtatja értékét, hanem esetleg a többi változó is előrébb/hátrább kerül a memóriában. Ezt a nehézséget úgy lehet kikerülni, hogy ha a gépi kódú programrészt tároló változó kialakítása előtt egyetlen más változónak sem adunk értéket. A meglehetősen bonyolult eljárás egyetlen előnye: a BASIC listán nem jelenik meg, s (idegen számára) nem is olyan könnyű megállapítani, hogy hol van a program, melyik változó rejti.
Láthatjuk tehát: a gépi kódú program helyének kiválasztása általában kompromisszumokkal jár. Tegyük fel, hogy döntöttünk, vagy kaptunk egy-két jó tanácsot, és rászántuk magunkat: a felsorolt lehetőségek közül melyiket válasszuk. De a lényeg hátra van: mi lesz maga a gépi kódú program?

3. Gépi kódú program beírása

Már említettük: a HT és a Sinclair gépek BASIC-ből nem értik a gépi kódot. De kihasználjuk, hogy a memóriának is, a Z80 mikroprocesszornak is mindegy, hogy amikor beleírunk, mire gondolunk: számra, karakterre, utasításra, tv-képre stb. Ha egy memória rekeszt azért olvas el a Z80, mert utasításkódot vár, akkor azt annak is fogja értelmezni. Ha tehát egy táblázatból rájövünk, hogy pl. a HT 1080Z BASIC-e az LD B,C gépi kódú utasítást, az A karaktert és a 65 egész számot (bizonyos feltételek esetén) egyformán tárolja, akkor a LD B,C utasítás memóriába írására akár a POKE. . ., 65 utasítás, akár az A karakter bevitele megfelel. Nézzünk ezekre egy-egy példát!
Balogh Györgyi idézett cikkéből világos, hogy ha a HT 1080Z gépen egy program első sora
10 REM . . . .
akkor a REM utáni 3. pont a 17 136 memóriacímen tárolódik, ha csak nem bűvészkedtünk előzőleg a géppel. Ezt bemutathatjuk egy POKE 17136,65 utasítással. Így egyben azt is látjuk: a 65 egész számot a POKE ugyanúgy kódolja, mint a BASIC interpreter az A betűt. Ha most még beetetnénk gépünket, és azt mondanánk neki, hogy a 17136 címen egy gépi kódú program kezdődik:
SYSTEM [NEW LINE]
/ 17136 [NEW LINE]
akkor számára ez nem 65-öt, nem A betűt, hanem egy LD B,C gépi kódú utasítást jelentene. De persze ennek a „gépi kódú program"-nak (micsoda nagyképűség!) se füle, se farka: ne csodálkozzunk tehát, ha hibajelzést kapunk.
Persze, ha nem 65-öt, hanem 201-et írunk, akkor már lehet némi reményünk: ez a gépi kódú RETURN utasítást. A SYSTEM-mel mégis megakadunk. (Aki akarja, erőltesse:

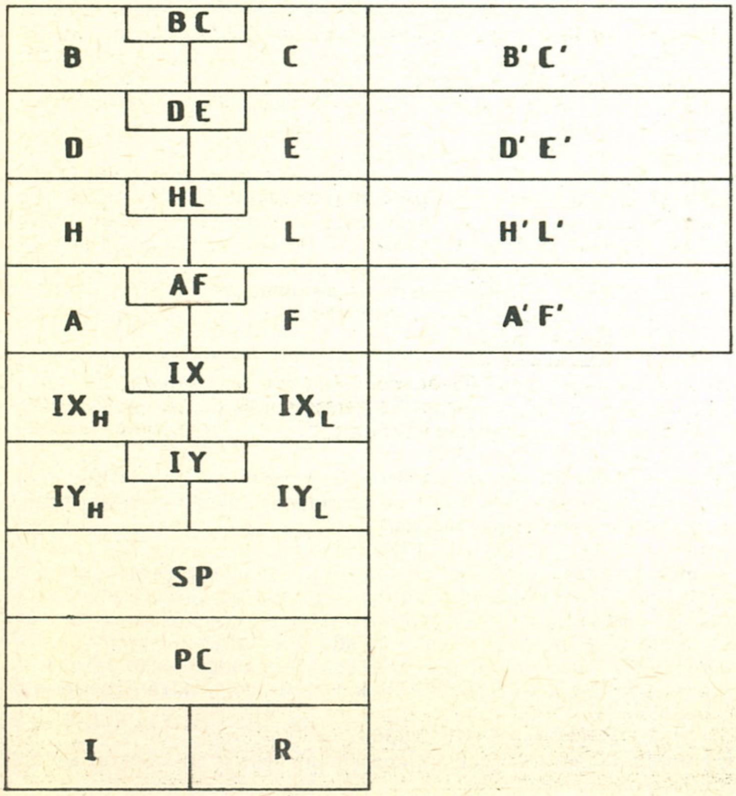
```
10 REM. . . . . . . . . . . . . . .
20 POKE 17136,201
30 POKE 16526,240
40 POKE 16527,66
50 Z = USR (0)
60 ? "KESZ"
```

is ugyanezt a gépi kódú rutint indítja, de most a RET jó helyre küld vissza.)
Ne essünk tévedésbe! Ami itt le van írva, az egy BASIC program, legfeljebb a benne levő POKE utasítások címei miatt a szokásosnál több ismeretet tételez fel. Az a gépi kódú rész, amiről szó volt: egyetlen-

egy byte, mégpedig a 17136-os memória-rekeszbeli gépi kódú RET. Ez a BASIC program írásakor még nem volt ott, a 20 jelű sor tette bele a RUN után. Nézzük meg, mi lett belőle (LIST)!
A 30 és 40 sor a 16526, 16527 rekeszekbe teszi a 17136 számot, 256-os számrendszerben (számoljunk utána: 240+256 • 66=17136). Azért ide, mert a BASIC interpreter részletes leírása szerint a BASIC program az USR függvény megtalálásakor ezen a helyen nézi meg, hogy hol kezdődik a gépi kódú programrész. Az 50. sor szerepe: a USR függvény indítása. Lehetne PRINT USR(0) is, csak a függvényekre vonatkozó helyesírást (szintaktikai szabályokat) kell betartanunk, a Z változónak semmi egyéb szerepe nincsen. A 60. sor a kísérletező kedvűek megnyugtatására szolgál: a gép és gépi kódú rész lefuttatása után visszatalált a BASIC programba, s folytatta annak végrehajtását (természetesen úgy, hogy egy bizonyos értéket beletett a Z változóba).
Érdemes ezt a primitív programot jól megérteni, mert más gépi kódú utasításokat „pókolva" a 10 REM . . . . -be, akármilyen (rövid) gépi kódú zsengénket futtatni tudja. Aki óvatos, vigye ki kazettára, még futtatás előtt. Így ha lemerevedik, meghal a rendszer kísérletezés közben, még ezt a pár sort sem kell ismét legépelni. Hosszabb programoknál ez feltétlenül megéri az időt.
ZX81 és Spectrum-használók is szeretnek az első REM-ben tárolni gépi kódú programot. A semmitmondó
10 REM . . . .
azonos lehet minden gépre, a memória-címet a kézikönyvek alapján meg lehet határozni. (Egy teljesen leírt programot találhat a kedves olvasónk a 97. oldalon.)

4. A CPU regiszterei

Ahhoz, hogy a gépi kódú utasításokat helyesen értelmezzük, a Z80 CPU (Central Processor Unit=központi feldolgozó egység) felépítéséről is tudnunk kell egyet-mást. Már említettük: van néhány (208) bit belső memóriája, ami számunkra hozzáférhető. Ez a belső memória általában 8 bites szervezésű, de egyes részei gyakran 16 bites regiszterekként működnek, néhány bit pedig csaknem minden utasítás szempontjából önálló életet él. Szokásos szakirodalmi megnevezésüket az ábra mutatja. Ugyanonnan leolvashatjuk a bitek számát is: az ábra a regisztereket jelző téglalapok méretével érzékelteti, hogy 8 vagy 16 bites regiszterről van szó. Pl. a B vagy az I 8 bites, a BC, SP, PC, A'F' stb. 16 bites.
Hangsúlyozni szeretném: a Z80 ezeket az elnevezéseket egyáltalán nem ismeri, kizárólag a gépi kódú utasításokat érti! Az itt használt jelöléseknek pl. a gépi kódú utasítások kikeresésekor van szerepe – akár magunk nézzük meg egy táblázatból, akár egy EDITOR-ASSEMBLER program keres a saját adataiból. Sokkal könnyebb az ember számára megjegyezni, hogy DEC H (decrement H=csökkentsd a H regiszterben levő számot eggyel), mint azt, hogy 37 (tizenhatos számrendszerben: 25). Pontosabban: egy-egy utasítás esetén talán ez nem így van – de gondoljuk meg: 0-tól 255-ig minden egész szám egy-egy utasítást is jelent, s ezeket mind „oda-vissza" kellene tudni, melyik milyen utasítást jelent. Ezért mi is használni fogjuk ezeket az elnevezéseket.
Az ábrán látható, hogy van pl. egy B jelű 8 bites regiszter. Erre elég sok utasítás vonatkozik, ugyanúgy a C jelű (szintén 8 bites) regiszterre. Van azonban néhány olyan utasítás is, amelynek végrehajtása során a BC regiszterpár együtt dolgozik, mint egy 16 bites regiszter. Ilyen együttműködési hajlamot mutat még D és E, H és L, IX_H és IX_L, IY_H és IY_L, és néha A és F. Más összekapcsolás (pl. A és H) nem lehetséges. Ezt a szoros kapcsolatot jeleztük az egyes regiszterpárok esetén az ábrán. Az IX és IY esetén meg kell mondani: szokás szerint ezeket mindig 16 bites regiszterekként használjuk, a ZILOG kiadású Z80 processzorok azonban értik a regiszterek felső, ill. alsó felére adott utasításokat is.
A regiszterek között van néhány speciális feladatot ellátó regiszter. Ilyen pl. az I, amelynek felhasználásával gyors, hatékony gépi kódú programot írhatunk az úgynevezett interrupt (megszakítás-kérés) jelek feldolgozására. Az IC két lába tud ilyen jeleket fogadni. Ezeket a jeleket küldheti egy másik egység, pl. egy printer vagy egy diszkmeghajtó, de jelezhet vészhelyzetet, tápfeszültség-kimaradást.
Az R regiszter a dinamikus RAM-ok (gyors, de nagyon feledékeny írható/olvasható memória IC-k) csatlakoztatását könnyíti meg: a felfrissítő (refresh) jelek képzésében játszik szerepet. (Csak a hardver iránt is érdeklődőknek: a dinamikus RAM-ok összes rekeszét működtetni kell, másodpercenként jó néhányszor, hogy ne sérüljön meg a bennük tárolt információ. A felfrissítő ciklus alatt egy rekeszt kiolvasnak, majd rögtön visszaírnak az erre tervezett áramkörök. Az R regiszter mutatja, hogy melyik rekesznél tart ez a felfrissítő tevékenység. Szerencsére ezt az egész feladatot ellátja a hardver, programozáskor nem is érezzük, ugyanis a CPU mindig olyankor ad frissítő jelet a RAM-oknak, amikor ő maga az utasítások dekódolásával van elfoglalva, tehát úgysem tudná értelmesen használni a külső memóriákat.)
A PC a már korábban is említett Program-Counter, program-lépésszámláló. Ez a regiszter tartalmazza annak a külső memóriarekesznek a címét, ahonnan a CPU a következő utasítást be fogja olvasni. Ha a CPU beolvas egy byte-ot, a PC értékét eggyel növeli – így az ismét a következő címre mutat. Még ha egy utasítás több byte hosszúságú, akkor is minden egyes byte beolvasása után automatikusan növekszik a PC-beli érték.
Az SP a Stack Pointer, a veremtár-mutató. Egy nagyon fontos adattárolási technikában jut szóhoz – de erről majd később.



GÉPI KÓDÚ UTASÍTÁSOK

Az LD-utasítások

Eddig csupán egyetlen gépi kódú utasítás, a RET szerepelt, melynek kódja 201 (=C9 a tizenhatos számrendszerben). REM sorba beírva a ZX 81-en TAN, a Spectrumon < > jelenik meg helyette.

Most rátérünk a Z80 mikroprocesszor egyik legnagyobb utasításcsaládjára, a LD utasításokra. Az LD a Load (=töltsd be) szó rövidítése. Több személyi számítógépen van ilyen feliratú billentyű – ahhoz azonban ennek a gépi kódú utasításnak lényegében semmi köze. A ZX 81 vagy Spectrum LOAD gombja a LOAD BASIC utasítást jelenti: tölts be magnóról egy programot. Itt, a LD gépi kódú utasításban szintén arról van szó, hogy valahonnan valamit be kell tölteni – de ez sohasem periféria, hanem egy külső vagy belső memóriarekesz. Emlékeztetünk arra, hogy a Z80 processzorban beépített regiszterek (memóriarekeszek) vannak, továbbá 64 K-s memóriát tud kezelni. Az LD utasítások segítségével lehet átvinni egy külső memóriarekesz tartalmát egy belsőbe vagy viszont, és ilyen utasításokkal oldható meg a belső regiszterek közti adatátadás. Végül: egy megadott értéket egy megadott külső vagy belső rekeszbe szintén LD utasításokkal írhatunk be. Nézzük részletesen! Kezdjük talán azzal, hogyan is tudunk megadott számot írni valamely külső vagy belső memóriarekeszbe!
A belső rekeszek közül az A, B, C, D, E, H, L regiszterekbe lehet közvetlenül betölteni egy megadott számot. Jele:
LD r,n
ahol r a fenti hét regiszter bármelyike lehet, n pedig 0–255 egész szám Pl. a C regiszterbe 71-et írni a
LD C, 71
utasítással lehet. Ezt ilyen formában persze csak mi értjük, meg az ASSEMBLER programunk (ha van), a Z80 nem. De rögtön számára is érthető lesz, ha

```
10 REM ................     Helyet biztosítunk a gépi kódú
                            programnak
20 POKE 16514,14            LD C,
30 POKE 16515,71                  71
40 POKE 16516,201           RET
50 PRINT USR 16514          A gépi kódú program indítása
                            BASIC-ből
```

formában közöljük vele egy ZX 81 gépen. A 16514 ZX 81 specialitás: a REM utáni első byte címe. HT 1080Z gépen 16514 stb. helyett 17134 stb. kell írni, és az 50 sortól:

```
50 POKE 16526,238 }         Felírjuk a gépi kódú program kez-
60 POKE 16527,66  }         dő címét az előírt rekeszekbe:
                            17134, 256-os számrendszerben.
70 PRINT USR (0)            A gépi kódú rutin indítása.
```

Megfigyelhetjük a LD r,n utasítás sajátosságait: a memóriában 2 byte-ot foglal el. Az elsőből kiderült, hogy LD utasításról van szó, sőt az is, hogy pont a C regiszterbe kell tölteni a következő memóriarekeszben található byte-ot. Ebből persze már az is következik, hogy a többi hat belső regiszterbe már nem a 14, hanem más-más kód tölt be! Itt és a hasonló jellegű utasításoknál az egyes regiszterek esetén a következő számokat kell venni:

A	B	C	D	E	H	L
7	0	1	2	3	4	5

Ezeket a számokat is r-rel jelölve, a LD r,n kódja

8r+6 ←első byte
n ←második byte

Így adódott LD C, n-re 8·1+6=14, és pl. LD B, n-re 6.
Feltűnő, hogy a 6 szám kimaradt. Pedig „él" a 8·6+6=54 kód! Jelentése:
LD (HL),n

Ez is LD utasítás, de ez nem belső regiszterbe, hanem a külső memória egy rekeszébe küldi az n-et – mégpedig oda, ahova a HL regiszterpár mutat. Ha pl. a HL tartalma 3C00=15360, akkor az n-et a 15360-as memóriarekeszbe írja be. Ezt a HT 1080Z gépen rögtön észrevesszük: a képernyő bal felső sarkában jelenik meg az n kódú grafikai jel vagy karakter (lásd: HT használati útmutató, 24. oldal: a tv-képernyőn megjelenítendő jeleket a 3C00–3FFF memóriarekeszek tartalmazzák). Ahhoz, hogy ezt kipróbálhassuk, természetesen előbb a HL regiszterpárba kell töltenünk a 15360-at. Meg tudjuk oldani:

```
38 }
60 } LD  H,60
46 }
 0 } LD  L,0
54 }
71 } LD  (HL), 71  ←G betű kódja
```

Az ellenőrizhetőség kedvéért a szükséges BASIC-sorokat is leírjuk:

```
15 POKE 17134,38   ←4·8+6    LD H,
20 POKE 17135,60                  60
25 POKE 17136,46   ←5·8+6    LD L,
30 POKE 17137,0                   0
35 POKE 17138,54   ←6·8+6    LD (HL),
40 POKE 17139,71                  71
45 POKE 17140,201  ←         RETURN
```

a 10 és az 50–70 sorok változatlanul maradnak. Vigyázzunk, a
10 REM ....
sorban legalább 7 db pont legyen!

A program indításakor a cursor nem állhat az alsó sorokban, mert akkor a READY megjelenésekor az automatikus SCROLL kitolja a képernyőről a G betűt. Talán legegyszerűbb a RUN parancs kiadása előtt törölni a képernyőt.

Ez a példa mutatja, hogy a LD r,n és a LD (HL),n utasítások alkalmazásával tetszőleges külső memóriarekeszbe beírhatjuk a kívánt byte-ot. Van azonban a fentinél egyszerűbb út is: a HL regisztertábra a 3C00 egyetlen utasítással is beírható:
LD HL, nn
melynek kódja 33. Az nn azt jelzi, hogy most két byte-ot kell megadnunk, hiszen a H és L regiszter összevontan is két byte-ot tartalmaz. A példánkhoz szükséges változat
LD HL, C300 vagy LD HL, 15360

Természetesen az ASSEMBLER-rel – és a tisztelt olvasóval – illene közölni, hogy mikor kell tízes, mikor tizenhatos számrendszerben érteni az LD-nél leírt számot. Az egyes assembler programok használati útmutatói leírják, hogy pl. $ vagy # stb. jelekkel jelezhetjük választásunkat. Itt a leírásban pedig a tizenhatos (=hexadecimális) számrendszert a szám mögé írt H jellel jelöljük. Hangsúlyozni kívánjuk azonban hogy akár
LD HL,3C00H akár LD HL, 15360
szerepel a leírásban, a memóriába kerülő gépi kódú változat azonos lesz!

Példánknál maradva:

Utasítás	Kódja tízes számrendszerben	Kódja tizenhatos számrendszerben	Karakter megfelelője HT, Spectrum	Karakter megfelelője ZX 81
LD HL,	33	21H	!	5
3C00H	0	00H		space
	60	3CH	<	W

Fontos megfigyelnünk, hogy a kétbyte-os adatot a Z80 úgy várja, hogy előbb az alacsonyabb, azután a magasabb helyiértékű byte-ot kell leírnunk. Ez a mikroprocesszor tervezőinek döntéséből következik, nem kell megérteni, csak következetesen betartani. Megjegyzését segítheti, hogy ily módon a programot tároló memóriában a nagyobb helyiértékű byte nagyobb címre kerül.

A kétbyte-os betöltő utasítás alkalmazásával 4 helyett csak 3 memóriarekeszre van szükségünk az utasítások tárolására. Az sem lényegtelen, hogy a kétbyte-os betöltő mintegy 3%-kal gyorsabb. Szerencsére nemcsak a HL, hanem még a BC, DE, SP, IX és IY regiszterekbe is betölthetünk egyszerre 2 byte-ot, az alábbiak szerint:

Utasítás	Kód hexadec.	Kód tízes
LD BC, nn	01H	1
LD DE, nn	11H	17
LD HL, nn	21H	33
LD SP, nn	31H	49
LD IX, nn	DDH 21H	221 33
LD IY, nn	FDH 21H	253 33

Például a 17 töltése az IX 16 bites regiszterbe:

LD IX, 17 = LD IX, 0011H

Kódolása: 221 33 17 0 tízes számrendszert használva
DDH 21H 11H 00H tizenhatos számrendszert használva

Mint látjuk, a 0 értékű „felső byte" (nagyobb helyiértékű byte) nem hagyható el: az LD IX,.. mindenképpen „lenyeli" a mögötte álló 2 byte-ot, azt a betöltendő 2 byte-ként értelmezve. Remélhetőleg a korábbi példák alapján világos: BC, DE, HL és SP esetében összesen 3 byte, IX és IY esetén 4 byte az utasítás helyigénye a memóriában. A betöltendő 2 byte az utasítás utolsó két byte-ja. Az IX-re ill. IY-ra vonatkozó utasítás a HL-re vonatkozóból úgy jött létre, hogy eléje került egy DDH ill. FDH „prefix" (=előválasztó).

Érdemes felfigyelni a

LD HL, nn (kódolása: 33 n_L n_H)

és a

LD (HL), n (kódolása: 54 n)

utasítások közti különbségekre. Így leírva ugyan elég nagy a hasonlóság, de jelentésük nagyon is eltérő. Ismétlésül:

LD HL, nn

a HL belső regiszterpárba tölti az nn kétbyte-os adatot;

LD (HL), n

pedig a HL által kijelölt külső memóriarekeszbe tölti az egybyte-os n adatot. A zárójel éppen azt jelenti, hogy nem a HL-re, hanem a HL által kijelölt helyre vonatkozik az utasítás. Ezt a jelöléstechnikát más esetekben is alkalmazza a Z80 leírása.

Ilyen például az

LD A, (nn) kódolása 58 n n
3AH

LD (nn), A kódolása: 50 n n
32H

egy byte-ot töltő és

LD rp, (nn) kódolása: 237 16·rp+75 n n
LD (nn), rp kódolása: 237 16·rp+67 n n

két byte-ot töltő utasítások, ahol rp a regiszterpár kódja:

0 1 2 3
BC DE HL SP

Jelentésüket a következő példák jelzik:

Ha pl. a
16396-os memóriarekesz tartalma 80,
16397-es memóriarekesz tartalma 70,
vagyis a 16396–397 memóriarekeszekben a
70·256+80=18 000
számot tároljuk, akkor az

LD A, (16396) az A akkumulátorba tölti a 16396 memóriarekesz tartalmát, a 80-at.

LD (16396), A a 16396-as memóriarekeszbe tölti az A akkumulátor tartalmát, a 80-at. (Áttöltés után az eredeti helyen változatlanul megmarad az érték, így inkább átmásolásról van szó!) Emlékeztetőül: a két utasítás konkrét alakjai:

Utasítás	Kód Decimális			Kód Hexadecimális		
LD A, (16396)	58	12	64	3AH	0CH	40H
LD (16396), A	50	12	64	32H	0CH	40H

hiszen 12+256·64 = 16396.

A fenti utasításokban A helyére nem kerülhet más belső regiszter! Az előbbi feltevések mellett

LD HL, (16396) (kódolása: 237 107 12 64 / EDH 6BH 0CH 40H)

a 16396–16397 memóriarekeszek tartalmát, 18000-at viszi a HL regiszterpárba,

LD (16396), HL (kódolása 237 99 12 64 / EDH 63H 0CH 40H)

a HL tartalmát viszi a 16396–16397 memóriarekeszekbe.

A jelölés következetes: a LD utáni első helyen mindig a „hova" kérdésre kapunk választ, a második helyen a „mit" kérdésre adódik a felelet. A zárójel szerepét, használatát gondosan figyeljük meg! A HL regiszterpár kitüntetett szerepű, vele kapcsolatban több olyan utasítás létezik, amelyeknek nincs közvetlen megfelelője a többi regiszterpárra. Egy ilyen különlegesség, hogy a fenti 4 byte-os utasítás helyett egy gyorsabb, 3 byte-os verzió is létezik

LD HL, (nn) kódolása 42 n n
ill.
LD (nn), HL kódolása 34 n n

értelemben. De ilyen rövidebb kódolása csak rp=HL esetben van, a többi regiszterpárnál az előbbi teljes verziót kell használni. Viszont HL esetében vétek lenne a memóriát és az időt pocsékolni. Egy jó ASSEMBLER mindig észreveszi, ha pont a HL-ről van szó.

A HL-re vonatkozó rövidebb alak lehetővé teszi, hogy az IX ill. IY-ra vonatkozó prefix alkalmazásával ezekre a regiszterekre szóló utasításokat írjunk:

LD IX,(nn)	kódolása	221	42	n	n
LD IY,(nn)		253	42	n	n
LD (nn), IX		221	34	n	n
LD (nn), IY		253	34	n	n

Kódoláskor mindig vigyázzunk az nn-et alkotó két byte helyes (fordított) sorrendjére!

Javasoljuk az Olvasóknak, hogy készítsenek 16×16-os táblázatot, és az eddig szereplő utasításokat írják be a hexadecimális kódjuknak megfelelő helyre. A prefixet tartalmazó (IX, ill. IY-ra vonatkozó ill. 237-tel kezdődő) utasításokat érdemes egy-egy külön táblázatba gyűjteni. Ha fóliára írjuk, a lapok egymásra helyezésével a prefixek hatását szemléltethetjük. De akár közönséges papírlapra írva is – nagyon jól látszik az egy lapra kerüűő utasításcsoportok logikája.

Az eddigi utasításokban mindig szerepelt egy egy- vagy két byte-os szám. A következő LD utasításokban ilyet nem találunk!

Az

LD r,r'

utasítások az r' regiszter tartalmát átmásolják az r regiszterbe. r és r' értéke egyaránt 0–7 lehet, ugyanúgy, mint az előzőekben:

A B C D E H L (HL)
7 0 1 2 3 4 5 6

Az utasítás kódolása:

8·r+r'+64,

tehát pl. LD C, D kódja 8·1+2+64=74, ez az utasítás a D regiszter tartalmát másolja át a C-be. Szabályos pl. a LD E,E utasítás is, de a LD(HL),(HL) verziót ne használjuk! Ez ugyanis az egyetlen kivétel: kódja, a 118, a HALT (állj meg!) utasítást jelenti: a mikroprocesszor felfüggeszti tevékenységét, amíg kívülről egy interrupt jelet nem kap. A többi variációval azonban szabadon kísérletezhetünk, a

LD(HL), r
és
LDr, (HL)

jelzésűeket is beleértve.

Az LD utasítások bő családjának végéhez közeledve megemlíthetjük még a következőket:

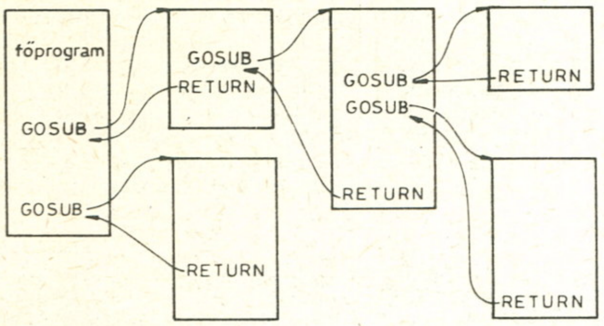
Egy byte-ot töltő utasítások:

LD A, (BC)	kódja:	10	ill. 0AH
LD (BC), A	kódja:	2	ill. 02H
LD A, (DE)	kódja:	26	ill. 1AH
LD (DE), A	kódja:	18	ill. 12H
LD I, A	kódja:	237 71	ill. EDH 47H

LD A, I kódja: 237 87 ill. EDH 57H
LD R, A kódja: 237 79 ill. EDH 4FH
LD A, R kódja: 237 95 ill. EDH 5FH
Két byte-ot töltő utasítások:
LD SP, HL kódja: 249 ill. F9H
LD SP, IX kódja: 221 249 ill. DDH F9H
LD SP, IY kódja: 253 249 ill. FDH F9H
További hat LD utasítást – formai okok miatt – az „Egyéb utasítások" között sorolunk fel.

PUSH és POP utasítások:

Rendkívül fontosak a
LD rp, (SP) és (POP)
LD (SP), rp (PUSH)
(rp: regiszterpár: AF, BC, DE, HL, IX, IY) típusú utasítások, de elnevezésük eltér az eddigiektől: az előbbieket POP, az utóbbiakat PUSH jelöli.
Működésüket érdemes részletesen követni.
A PUSH és a POP utasítások segítségével kényelmesen kezelhetjük az úgynevezett stack-memóriát: a PUSH az adatok beírására, a POP a kiolvasásra való. A stack nem egy újabb memória-gyártmány, hanem az adattárolás egy lehetséges megszervezését jelenti. Ami a szó eredeti jelentését illeti, az angol szótárban „stack: boglya, kazal, asztag, (fa)rakás" található. Mire is gondoljunk most?
Amikor a BASIC-ben szubrutint hívunk (GOSUB), az interpreter (=a BASIC-et értelmező program) köteles megjegyezni ennek helyét, hogy ha majd RETURN utasításhoz ér, a megjegyzett helytől (a GOSUB utáni első utasítástól) tudja folytatni programunk futtatását. Ezt úgy modellezhetjük, hogy a megjegyzendő címet felírjuk egy cédulára. Ha a program futása során újabb GOSUB-hoz érünk, mielőtt RETURN-re lépnénk, akkor egy újabb cédulát helyezünk az előzőre, s a felsőre ráírjuk az újabb visszatérési címet – most már csak ez látható. Ha viszont RETURN utasításhoz érünk, elolvassuk a legfelső visszatérési címet, majd eldobjuk a legfelső cédulát. Így akár több száz RETURN-nel le nem zárt szubrutinhívás is megjegyezhető, csak közben elég magasra nő a cédularakás. Így már érthető a stack elnevezés – és azon sem lepődünk meg különösebben, hogy magyarul veremtárról szokás beszélni: a ez szókép azt is visszatükrözi, hogy a verem előbb-utóbb betelik, ha mindig csak belerakjuk a cédulákat anélkül, hogy ki is vennénk.
A stack (veremtár) esetében nagyon fontos, hogy nem a legrégebbi, hanem ellenkezőleg, a legújabb cédula az, amit először elolvasunk!
A RETURN-nél ez a BASIC definíciója szerint szükséges: először mindig a legutoljára megkezdett szubrutin fejeződik be (mint a zárójeleknél!).



A RETURN utasítások egy része szándékosan áll a szubrutint jelző téglalap belsejében, ezzel is hangsúlyozva, hogy a RETURN csak a végrehajtás sorrendjében az utolsó utasítás a szubrutinon belül, helyileg állhatnak mögötte programsorok, és pl. egy kisebb sorszámra mutató GOTO lehet a legalsó programsor. (Gondoljuk meg, mit jelent, ha a szubrutin utolsó sora STOP vagy END!)
Természetesen nem csak szubrutinhívással kapcsolatos visszatérési címeket, hanem tetszőleges egyéb adatokat is beírhatunk a stackterületre – csak arról nem szabad elfelejtkezni, hogy kiolvasáskor fordított sorrendben tudjuk azokat elérni. A PUSH és POP utasítások pontos alakja és kódjaik a következők:

Utasítás	Kód Tízes	Kód Hexa	Utasítás	Kód Tízes	Kód Hexa
PUSH AF	245	F5	POP AF	241	F1
PUSH BC	197	C5	POP BC	193	C1
PUSH DE	213	D5	POP DE	209	D1
PUSH HL	229	E5	POP HL	225	E1
PUSH IX	221 229	DD E5	POP IX	221 225	DD E1
PUSH IY	253 229	FD E5	POP IY	253 225	FD E1

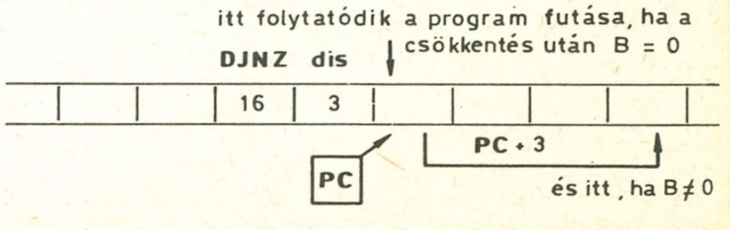
Példaként a PUSH HL és a POP HL utasítás működését írjuk le (SP a stackpointer):

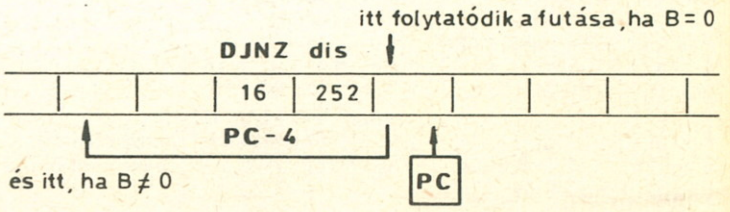
PUSH:	POP:
1. SP=SP−1	1. LD L, (SP)
2. LD (SP), H	2. SP=SP+1
3. SP=SP−1	3. LD H, (SP)
4. LD (SP), L	4. SP=SP+1

A négy ütem lefutása mindkét esetben automatikus! Maga az utasításkód egyetlen byte – viszont a végrehajtáskor két byte-ot másol át a stackpointer által mutatott helyre ill. helyről kezdve. Továbbá megfelelően módosítja a stackpointer értékét is, hogy az készen álljon a következő PUSH vagy POP vezérlésére. Éppen az SP regiszter módosítása miatt a PUSH és a POP nem tisztán LD típusú utasítás, hanem annál összetettebb.
Ha nem a HL regiszterpárra vonatkozik a PUSH ill. POP, akkor H helyére a magasabb, az L helyére az alacsonyabb helyiértékű regiszter lép; AF esetén A számít a magasabbnak.
A PUSH és a POP utasítások segítségével hozzáférhető lesz az F regiszter is, amelyre nincs is más LD típusú utasítás. PUSH AF; POP BC után F tartalma C-be kerül, míg PUSH BC; POP AF hatására C tartalma betöltődik F-be.
Megjegyezzük még, hogy mindegyik említett személyi számítógép használ veremtárat, ezért az SP veremtármutató már a BASIC feléledésekor megfelelő kezdőértékkel rendelkezik. Így minden különösebb előkészület nélkül használhatjuk a PUSH és POP utasításokat, csak arra kell vigyázni, hogy a memória be ne teljen. Természetesen az SP át is állítható nekünk tetsző területre, pl. az LD SP, . . . utasítások valamelyikével –, ilyenkor célszerű az eredeti értéket elmenteni, s a gépi kódú rész befejezése előtt visszatölteni, hogy a BASIC-be visszatérve folytatódhasson a program futása.

DJNZ — ciklusszervezés:

Most megismerkedünk egy nagyon hasznos utasítással, amelynek segítségével könnyen lehet ciklust szervezni. Ez egy speciális feltételes ugró utasítás, jele
DJNZ dis
ahol dis egy byte-ot jelent, amely megmutatja, hogy a feltételek teljesülése esetén hová kell ugrani. (DJNZ=Decrement B+Jump if non zero; csökkentsd a B regiszter tartalmát és ugorj, ha nem zéró, dis=displacement: elmozdítás.) Maga az utasítás két byte-ot foglal el a memóriában: az első a kód, 16 (=10H), ez jelzi, hogy DJNZ utasításról van szó; a második byte a dis érték tárolására szolgál (hogy hogyan, arra hamarosan visszatérünk).
Hogyan működik a DJNZ utasítás? Természetesen először is magát az utasítást olvassa el a CPU, mind a két byte-ot, tehát a PC (program lépésszámláló), a dis után következő byte-ra mutat. Az utasítás végrehajtásakor a CPU először is eggyel csökkenti a B regiszter tartalmát, majd ezután megvizsgálja, hogy a B nulla-e vagy sem. Ha a B=0, akkor a PC értéke nem változik, ezért a program futása a dis utáni byte-on folytatódik. Ha azonban B≠0 a csökkentés után, akkor a PC tartalmához a CPU hozzáadja a dis úgynevezett kettes komplemensét. Ez azt jelenti, ha
$00 \leqslant dis \leqslant 127$, akkor
PC=PC+dis,
ha pedig $128 \leqslant dis \leqslant 255$, akkor
PC=PC+dis−256
vagyis a dis legnagyobb helyiértékű bitje szabja meg, hogy a PC értéke növekszik vagy csökken.
A képlet azt jelzi, hogy ha dis helyére 1–127 kerül, akkor előre ugrunk a programban, ha pedig 128–255, akkor visszafelé, a dis utáni első byte-hoz képest.
Nézzünk két példát, amely gépi kódú programot tartalmazó memóriarészletet szemléltet:





A most következő ábráról leolvasható, hogy milyen számot kell írni a dis helyére, ha csak néhány lépésnyit akarunk előre vagy hátra ugrani:

			DJNZ	dis						
251	252	253	254	255	0	1	2	3	4	5
FB	FC	FD	FE	FF						
-5	-4	-3	-2	-1						

Hogy az ábra alá írt három sor közül melyiket kell használni, az attól függ, hogy milyen EDITOR-ASSEMBLER programunk van. Természetesen az a legkényelmesebb, ha az editor megengedi címke használatát, s maga számítja ki a dis értékét. Ha gépi kódú programunkat POKE utasításokkal írjuk be, mindig a 0–255 számokat kell használnunk. (Esetleg, ha megengedett, negatív számokat.) A DNJZ utasítással kapcsolatban tudnunk kell, hogy ha B=0 a DJNZ végrehajtása, tehát már a csökkentés előtt, akkor az első ütemben B=255 lesz (a „kivonás" eredményeként), tehát B=0, vagyis a program a disnek megfelelő helyen fog folytatódni.
Nézzük meg, hogyan lehet ciklust szervezni a DJNZ utasítás segítségével!
Először is a ciklusmag (az ismétlődő rész) elé egy
LD B, n
utasítást írunk, ahol n-től függ, hogy hányszor fut el a ciklus. Itt n értéke 0–255 egész szám lehet. A ciklusmag mögé kerül a DJNZ utasítás:

```
LD B,n
.
.
.
.
.
.
.
.
DJNZ dis
```

ahol dis értékét úgy kell megválasztani, hogy B=0 esetén a ciklusmag első byte-járól folytatódjon a program futása. Ha például a ciklusmag 5 byte hosszúságú, akkor DJNZ –7 (=DJNZ 249) lesz a megfelelő, hiszen a PC már a dis utáni byte-ra mutatott, s ezért a „DJNZ" és a „dis" byte-okat is át kell ugrani visszafelé.
Ha az n értéke 1–255, akkor a ciklusmag n-szer fut le. Ha azonban n=0, akkor a DJNZ utasítás már említett sajátossága miatt az első lefutáskor B értéke 255 lesz, azaz nemcsak, hogy nem fejeződik be azonnal a ciklusmag futtatása, hanem még 255-ször, összesen tehát 256-szor kerül sor a ciklusmag végrehajtására!
Megemlítjük, hogy

```
LD B, n
DJNZ –2
```

n-szer lefutó üres ciklus, hiszen DJNZ –2 (=DJNZ 254) magára a DJNZ utasításra ugrat). Ha n> 0, akkor ez a
FOR I = n TO —1 STEP –1: NEXT I
BASIC-utasításnak felel meg, amit (inkább
FOR I=1 TO n: NEXT I
alakban) gyakran felhasználunk várakozásra, a program lassítására.
Most hozzákezdünk az aritmetikai és logikai műveletekhez. Ezekből is van jó néhány. Ennek ellenére: még szorozni sem tud a CPU! Hát akkor hogyan működnek a BASIC-ben az alapműveletek? Úgy, hogy az ezután következő elemi műveletek segítségével megírták mindegyiknek a programját! Nem hangzik valami biztatóan – de ha már hozzákezdtünk a gépi kódhoz, folytassuk!

Aritmetikai és logikai műveletek

1. 16 bites INC és DEC utasítások

Először is az INC művelettel ismerkedünk meg. Ez a jelölés az increment (= növekedés) szó rövidítése; itt pontosabban az 1-gyel való növelésről lesz szó. INC HL tehát azt jelenti, hogy a HL regiszterpár tartalmához (mint 16 biten ábrázolt kettes számrendszerbeli pozitív egész számhoz) 1-et ad a CPU. Az INC párja a DEC (decrement = csökkentés): DEC HL regiszterpár értékét 1-gyel csökkenti. A 16 bites regiszterek közül a következőkre működnek:

Művelet	Kódja Tízes	Kódja Hexa	Művelet	Kódja Tízes	Kódja Hexa
INC BC	3	03 H	DEC BC	11	0BH
INC DE	19	13 H	DEC DE	27	1BH
INC SP	51	33 H	DEC SP	59	3BH
INC HL	35	23 H	DEC HL	43	2BH
INC IX	221 35	DD 23 H	DEC IX	221 43	DD 2BH
INC IY	253 35	FD 23 H	DEC IY	253 43	FD 2BH

Mintaprogram
Most egy példával szemléltetjük az INC HL működését. Vegyük alapul a következő assembler programot:

```
LD HL, kép
LD A, kód
LD (HL), A
RET
```

Ha „kép" a display (képernyő) memória egy helyének címe, „kód" pedig egy képernyőn megjeleníthető jel kódja, akkor programunk a „kép" címre teszi a „kód" jelet, vagyis megfelel a
POKE kép, kód
BASIC utasításnak. Ha most a fenti helyett a

```
LD HL, kép
LD A, kód
LD (HL), A
INC HL
LD (HL),A
RET
```

programot futtatjuk, akkor a „kép" és a „kép"+1 címen is megkapjuk a „kód"-nak megfelelő jelet. Ha többször megismételnénk a most beszúrt utasításpárt, még több jelet kapnánk a képernyőn, egymás után. Ezt azonban meg ne tegyük a világért se! Erre való a DJNZ. Nézzük csak:

Utasítás	Kód decimális	Kód hexadecimális
LD HL, kép		
LD A, kód		
LD B, n		
LD (HL), A	119	77H
INC HL	35	23H
DJNZ –4	16	10H
	252	FCH
RET	201	C9H

A kódokat csak a négy utóbbi utasítás esetén írtuk ki, ez is elég a DJNZ működésének megértéséhez. A DJNZ végrehajtásakor a PC a 201-es kódú RET (return) utasításra mutat. Ha B=0, akkor innen 4-et lép visszafelé, vagyis ismét az LD (HL),A utasítás kerül sorra. Igenám, de az első lefutás óta már volt egy INC HL, vagyis a HL tartalma most eggyel több, mint előzőleg, ezért most az A-ban őrzött kód a kép+1 pozícióba kerül. A ciklusmag az LD (HL),A és INC HL utasításokból áll, ezek ismétlődnek n-szer. De a HL mindig eggyel előrébb mutat, s így n db egymás utáni helyen jelenik meg a képernyőn a jel.

Itt az ideje, hogy elméletileg vizsgált programunkat futtassuk. Meg is tesszük, mégpedig a HT 1080Z, PRIMO, Spectrum és a ZX 81+16 K RAM esetére közöljük a teljes programot. A négy gépen a betöltés és a gépi kódú rész négy különböző változatot jelent, mutatva, hogy a gépek program- és képtárolása eltérő. Mégis megpróbáltuk a közös elemeket hangsúlyozni, s olyan programot adni, amelyik minimális változtatással mind a négy gépen futtatható. Ne csodálkozzanak hát az egyes gépek jó ismerői, ha egyszerűsíteni tudják a most következő programot: az „ügyetlenkedés" hol egyik, hol másik gép kedvéért történik. Ezek után lássuk a BASIC programot:

```
10 REM . . . . . . . . .
20
30 FOR I = CIM TO CIM+12
40 PRINT I,
50 INPUT X
60 POKE I,X
70 NEXT I
80 CLS
90 PRINT
100 PRINT
110
120 PRINT "KESZ"
```

A 20-as és 110-es sort gépenként írjuk le:

HT 1080Z és PRIMO
20 L = PEEK (16548):H = PEEK (16549):CIM = 256*H+L+5
SPECTRUM
20 L = PEEK 23635:H = PEEK 23636:CIM = 256*H+L+5
ZX 81 + 16 K RAM
20 CIM=16514
HT 1080Z
110 POKE 16526, L:POKE 16527, H:PRINT USR (0)
ZX 81 + 16 K RAM és SPECTRUM
110 PRINT USR CIM
PRIMO
110 PRINT CALL (CIM)

MA BIZTOSÍTJUK A HOLNAP TECHNIKÁJÁT!

A MŰSZERTECHNIKAI KISSZÖVETKEZET AZ ALÁBBI MIKROSZÁMÍTÓGÉP TÍPUSAIT AJÁNLJA AZ ÉRDEKLŐDŐK FIGYELMÉBE:

MULTI CENTER

- 8 MUNKAHELY • 256 K RAM • HÁTTÉRTÁR WINCHESTER

MXT

- IBM XT KOMPATIBILIS SZÁMÍTÓGÉP

MXT–M

- IBM XT KOMPATIBILIS MULTITERMINÁLOS SZÁMÍTÓGÉP

MULTI WS

- VME BUS KOMPATIBILIS TÖBB PROCESSZOROS, TÖBB MUNKAHELYES CAD/CAM RENDSZER

TRANSMIC–8

- HORDOZHATÓ SZÁMÍTÓGÉP

TZ–80

- ASZTALI SZÁMÍTÓGÉP • STANDARD OPERÁCIÓS RENDSZEREK ÉS PROGRAMCSOMAGOK HASZNÁLATA.
- VÁLLALJUK KULCSRAKÉSZ RENDSZEREK SZÁLLÍTÁSÁT EGYEDI IGÉNYEK SZERINT

SZÁMÍTÓGÉPEINKET, SZOFTVEREINKET MEGTEKINTHETIK, MEGVÁSÁROLHATJÁK, LÍZINGELHETIK COMPUTER BOLT-UNKBAN.
1075 BUDAPEST, MAJAKOVSZKIJ U. 1/D. TEL.: 423-423
NYITVA: H–P. 9–17 ÓRÁIG.

Futtatáskor ezek a BASIC programok „pókolják" be a gépekbe a gépi kódú programunkat. Mindegyikük tízes számrendszerben kéri a gépi kódú utasítások kódját. A 10–70 sorok egy általános érvényű betöltő programot alkotnak: más gépi kódú program esetén csak a 30 sor végén álló 12 számot kell kicserélni a betöltendő program hosszánál eggyel kevesebbre (mi most egy 13 byte-os programot töltünk be, ezért szerepel 12), és a 10 sorba kell elegendő pontot írni. A CIM változó a 10 REM utasítás első pontjának címe, innen kezdve töltődik be a gépi kódú program, amelynek indítása (futtatása) mindegyik gép esetén a 110-ből történik. A 120 szemlélteti: a gépi kódú program lefutott, s a vezérlés szerencsésen visszakerült a BASIC programba. A 80, 90 és 100 sorok általában feleslegesek, de most a betöltendő programunk a képernyő első 1–2 sorába ír – ehhez biztosítanak helyet a képernyőn.
A betöltő programokat RUN-nal indítva 13 számot kérnek inputként. Az első három a géptől függ:

	HT 1080Z	PRIMO	Spectrum	ZX 81+ 16K RAM
1.	33	33	33	42
2.	1	1	1	12
3.	60	232	64	64

A további tíz szám már közös: 35, 62, 33, 6, 16, 119, 35, 16, 252, 201.
Vigyázat: a futtatás megkezdése után már mindenféle zűrökre számíthatunk (pl. a 10 REM... többé már nem listázható, s a SAVE [CSAVE] paranccsal is gondjaink lehetnek). Ezért ajánlatos még az első RUN előtt kazettára menteni a betöltő BASIC programot.
Mit fogunk látni, ha mindent hibátlanul végeztünk el? A 9 utolsó utasítást visszafejtve, a gépi kódú rész jelentése: LD A,33:LD B, 16:LD(HL),A:INC HL: DJNZ –4:RET vagyis 16 db 33-as kódú jel látható a képernyő legfelső sorában. Azért ott, mert az első négy byte oda állította be HL kezdő értékét. LD HL,nn : INC HL, ill. ZX 81-nél LD HL, (nn):INC HL.
A kialakuló HL érték:

	HT 1080 Z	PRIMO	SPECTRUM	ZX 81
HL	15362	59394	16386	D–FILE+1

ahol D-FILE a 16396, 16397 memóriarekeszekben tárolt cím. Ezeket a számokat az egyes gépek leírása alapján választottuk ki.
GOTO 80-nal mindegyik gépen újra futtatható a gépi kódú program, nem kell még egyszer betölteni. LIST 20, ill. LIST 20– hatására az első sor kivételével listázhatók, sőt, ZX 81-nél a teljes lista olvasható.
A POKE CIM+5, kód
vagy
POKE CIM + 7, db
után
GOTO 80
hatására a kód-nak megfelelő db számú jelet kapjuk. Ez HT 1080 Z és ZX 81 esetén a karakterkódra vonatkozik. (33 a felkiáltójel, ill. az 5 számjegy kódja), a Spectrum és a PRIMO esetében pedig a finom grafika szabályainak megfelelő pontsorozathoz jutunk. (33 = 32+1, tehát byte-onként 2 pont, a 32 és az 1 helyiértékű pozíciókban.)
Ha a kód és db értéket többször is változtatni akarjuk
25 GOTO 80
beszúrása után egyszerűen RUN-nal indíthatjuk programunkat. Vigyázzunk arra, hogy a ZX81-nél 16 KRAM esetén a D-FILE címen, továbbá a sorok végén kötelezően NEW LINE karakter áll: oda tehát ne írjunk semmilyen más jelet! (Írhatunk – majd meglátjuk, hogy eltorzul a kép – kikapcsolás nélkül aligha ússzuk meg!). Itt jegyezzük meg, hogy a 4.byte-ként írt 35, azaz INC HL kizárólag a ZX 81 számára volt fontos, a többi gépnél csak az egyformaság kedvéért szerepelt, hiszen nem zavarta a működést. Így az első három gépnél megtehetjük, hogy a tíz „közös" lépésként az alábbiakat adjuk meg: 62, 33, 6, 16, 119, 35, 60, 16, 251, 201.
(ZX 81-nél a betöltő programban írjunk 12 helyett 13-at) s akkor az eredeti 4 első byte után befér a mostani tíz is). Most már különböző jeleket kapunk a képernyőn! Ennek oka természetesen a 60-as kódú utasítás beszúrása, melynek jelentése: INC A, vagyis: növeli meg eggyel az A regiszter tartalmát!
Mielőtt visszatérnénk a Z80 utasítások ismertetéséhez megjegyezzük, hogy a betöltő BASIC programok
10 REM ....
20 (mint az előző változatban)
30 POKE CIM, 33
40 POKE CIM +1,1
50 POKE CIM +2,60
.
.
.
módon is átírhatók. Így kicsit többet kell gépelnünk, de ismételt RUN-ra sem lesz probléma, és a SAVE (CSAVE) menti a gépi kódú program-kódjait is. Ez persze megoldható READ/DATA technikával, ahol van. Érdekességként megemlítjük, hogy a PRIMO gépen egyszerűen
10 REM ....
20 (mint az előzőekben)
30 POKE CIM, 33, 1, 232, 62, 33, 6, 16, 119, 35, 60, 16, 251, 201
40 CLS: PRINT: PRINT: PRINT CALL (CIM) "KESZ"
írható és futtatható.

2. 8 bites INC, DEC utasítások

A 16 bites INC és DEC utasítások után most a 8 bites INC és DEC utasításokat ismertetjük:

Utasítás:	Kódja:	C	Z	P/V	S	N	H
INC r	8·r+4	·	↕	V	↕	0	↕
INC (IX+d)	DDH 34 H ill. 221 52	·	↕	V	↕	0	↕
INC (IY+d)	FD H 34 H ill. 253 52	·	↕	V	↕	0	↕
DEC r	8·r+5	·	↕	V	↕	1	↕
DEC (IX+d)	DDH 35 H ill. 221 53	·	↕	V	↕	1	↕
DEC (IY+d)	FD H 35 H ill. 253 53	·	↕	V	↕	1	↕

(Flagek: C Z P/V S N H)

Az r értéke 0–7 lehet, az LD utasításoknál megszokott módon, vagyis az egyes regiszterek számozása:
A B C D E H L (HL)
7 0 1 2 3 4 5 6
Tehát INC C kódja 8·1+4=12, DEC (HL) kódja 8·6+5=53 a tízes számrendszerben. Emlékeztetünk rá:
DEC HL a HL regiszterpárban tárolt 16 bites számot csökkenti eggyel;
DEC (HL) a HL regiszterpárban tárolt számot változatlanul hagyja, hanem ezt a számot mint memóriacímet értelmezve az illető memóriarekesz tartalmát csökkenti eggyel.
A HL-re vonatkozó utasításkód elé kiírva a 221, ill. 253 utasításkódot az eddigi szokásnak megfelelően, IX-re ill. IY-ra vonatkozó utasítást kapunk. Ezek az utasítások azonban nem 2, hanem 3 byte-osok, pl.: INC (IX+d), ha d=C2H, akkor a programtárban:
DDH 34H C2H, vagyis
221 52 194
alakban tárolható. A d a DJNZ kapcsán megismert dis szerepét játssza: a processzor az IX regiszter tartalmának változatlanul hagyásával kiszámítja az IX+d összeget, majd ezt memóriacímként értelmezve növeli a szóban forgó memóriarekesz tartalmát.

FLAGEK:
A kódok felsorolásánál újdonság a „Flagek" rovat. Az előzőekben már említettük, (93. old.), hogy a Z80 processzornak van egy F jelű regisztere. A PUSH és POP utasításoknál (96. old.) szó volt az F regiszter eléréséről. Mostani utasításainkban azonban nagyon fontos szerephez jut az F regiszter: az INC végrehajtása során nemcsak az utasításban kijelölt regiszter ill. memóriarekesz tartalma változik, hanem az F egyes bitjei is. Ezeket a biteket mint jelzőket (flag= zászló) használják egyes utasítások. Korábban azért nem emlegettük, mert – a 16 bites INC és DEC utasításokat is beleértve – az eddig ismertetett utasítások nem módosították az F bitjeit (szigorúan nézve: az LD A, I és az LD A, R kivételek, módosítják a flageket).
Az F regiszter 8 bitje közül 6 bit kapott nevet a Z 80 leírásban.

7	6	5	4	3	2	1	0
S	Z	☒	H	☒	P/V	N	C

S: Sign (előjel)
Z: Zero (nulla)
H: Half carry (BCD átvitel)
P/V: Parity / Overflow (paritás/túlcsordulás)
N: kivonás
C. Carry (átvitel)
Két bit funkcióját az eredeti INTEL leírás nem közli.
Az utasításkódok táblázatában a · azt jelenti, hogy az illető bit nem változik a művelet elvégzése során, ↕ jelöli, hogy a bit értéke a művelet végeredményétől függ, 0 ill. 1 pedig azt, hogy a bit értéke a végeredménytől függetlenül 0, ill. 1 lesz.
A P/V oszlopban a V arra utal, hogy ez a bit a művelet végzése során túlcsordulásjelző szerepet tölt be. Ennek jelentéséről, valamint egyéb előforduló jelölésekről később lesz szó (l. 102. old.).
Az egyes flagek jelentését, felhasznál[illegible]tekben ismertetjük.

A flagek szerepe majd akkor lesz világos, ha értéküket fel tudjuk használni (pl. feltételes ugró utasításokban). A legegyszerűbb az
S flag:
Az S egyszerűen a végeredmény legmagasabb helyiértékű (7.) bitjének másolata.
Ugyancsak egyszerű a
Z flag:
Ez jelzi, hogy ha a végeredmény 0. Ezt úgy kell érteni, hogy ha a végeredmény 0, a Z flag értéke 1 lesz, egyébként 0. Így pl. ha az INC vagy DEC művelet végeredménye 0 lesz, akkor a Z bit 1 lesz, egyébként a Z bit 0. Ne csodálkozzunk: az INC eredménye is lehet 0: ugyanis 255+1 elvégzésekor az eredmény 0, és a Z bit 1 lesz.

3. Összeadó és kivonó utasítások

A következő utasítás, amivel megismerkedünk, a 8 bites összeadás:

Utasítás:	Kódja:	Flagek: C	Z	P/V	S	N	H
ADD A, r	128+r	↕	↕	V	↕	0	↕
ADD A, (IX+d)	DDH 86H ill. 221 134	↕	↕	V	↕	0	↕
ADD A, (IY+d)	FDH 86H ill. 253 134	↕	↕	V	↕	0	↕
ADD A, n	C6H ill. 198	↕	↕	V	↕	0	↕

Az utasítások kódolásánál ne felejtkezzünk meg arról, hogy az (IX+d) ill. (IY+d) típusúak esetében a kód 3 byte-ot foglal el a memóriában: az első kettő a táblázatban szereplő 2 byte, a harmadik a d értéke (1 byte).
Az ADD A, n forma kódolása két byte-on történik, a második n értéke: amennyit az A regiszter tartalmához hozzá kell adni. A végeredmény minden esetben az A regiszterbe kerül.
Ha pl. A tartalma 42
C tartalma 57
akkor a 129 kódú ADD A, C hatására A tartalma 99 lesz, C változatlanul 57 marad.
De mi lesz a 112+107 vagy a
112+157
összeadás eredménye?
Kísérletezésre kapható Olvasóinknak egy kis BASIC programot ajánlunk a HT 1080 Z-re kidolgozott változatban. (Írták: Brányi László és Szenttornyai László.)
A BASIC program megírja a szükséges gépi kódú betétet, ezért gond nélkül (CSAVE "G") save-elhető. (Vigyázzunk, hogy még a próbafuttatás előtt vigyük ki kazettára, mert később már gondok adódhatnak! Ugyanis, ha a kialakuló gépi kódú programban 0 van, akkor azt a BASIC interpreter sorvége jelnek tekinti, s onnan kezdve új BASIC sorként kívánja értelmezni a soron következő byte-okat, ami természetesen nem jár sikerrel.)

```
10 REMAB+F...................19 DB PONT
20 DIMB$(7)
30 DIMC$(7)
40 DATA"SIGNUM (ELOJEL)","ADD A,","ZERO ","ADC A,","NEM HASZNALT","SUB "
41 DATA"HALF CARRY","SBC A,","NEM HASZNALT","AND ","PARITY/OVERFLOW"
42 DATA"XOR","NEG.","OR","CARRY","CP"
50 DATA237,75,241,66,197,241,33,238,66,126,35,134,245,193,35,119,35,113,201
60 FORN=7TO0STEP-1
70 READ B$(N),C$(N)
80 NEXT
90 FOR N=17138TO17156
100 READ P
110 POKE N,P
120 NEXT
130 CLS
140 FORN=0TO7
150 PRINTN;"    ";C$(7-N)+"(HL)"
160 NEXT
170 A$=INKEY$
180 IFA$<"0"ORA$>"7"THEN170
190 POKE253+66*256,134+VAL(A$)*8
200 F=0
210 X$=""
220 FORN=7TO0STEP-1
230 PRINTB$(N);"ERTEKE?";
240 A$=INKEY$
250 IFA$<"0"ORA$>"1"THEN240
260 X$=X$+A$
270 PRINTA$
280 F=F+2↑N*VAL(A$)
290 NEXT
300 POKE241+66*256,F
310 INPUT"'A' ERTEKE";A
320 IFA<0ORA>255T310
330 POKE17134,A
340 INPUT"'B' ERTEKE";B
350 IFB<0ORB>255T340
360 POKE17135,B
370 POKE16526,242
380 POKE16527,66
390 C=USR(0)
400 PRINT"AZ EREDMENY=";PEEK(17136)
410 PRINT,"UJ","REGI"
420 F=PEEK(17137)
430 FORN=7TO0STEP-1
440 PRINTB$(N),
450 IFF>=2↑NTF=F-2↑N:PRINT": 1",ELSEPRINT": 0"
460 PRINTMID$(X$,8-N,1)
470 NEXT
480 IFINKEY$=""THEN480ELSE140

READY.
```

Indításkor a program megkérdi, hogy 8 általa ismert művelet közül melyiket szemléltesse, majd kér két egybyte-os számot (0–255) tízes számrendszerben, megkérdi, hogy az egyes flagek értéke mi legyen a művelet elvégzése előtt, majd elvégzi a műveleteket, leírja az akkumulátor és a flagek tartalmát. A lehetséges műveletek egy byte-ra vonatkozó utasítások: ADD, ADC, SUB, SBC, AND, XOR, OR, CP, (HL)-re vonatkozó alakjai. Most még csak az ADD utasítással foglalkozzunk. Javasoljuk, hogy jól figyeljék meg, milyen összeadások eredményénél lesz a C ill. P/V flag értéke 1. (Vigyázat! a C regiszternek nincs semmi köze a C flaghez, csak a nevük hasonló!)
A flagek működését bemutató program használata során hamar feltűnik: a két „nem használt" flag is dolgozik (pl. 253+2, 125+99, 125+12), csak feladatukat nem részletezi a gyári leírás. Feltételes utasítás sem tartozik ehhez a két bithez. A CARRY-flag (az F regiszter legkisebb helyiértékű bitje) akkor lesz 1, ha a két összeadandó összege legalább 256. Az úgynevezett több byte-os számábrázolásnál van fontos szerepe.
Ha nem elégszünk meg a 0–255 egészek ábrázolásával, amire egy byte alkalmas (és miért is elégednénk meg?), akkor több byte-on tárolunk egy-egy számot. Ilyenkor az egyes byte-ok szerepe hasonló a tízes számrendszerbeli számjegyekéhez: egymás mellé írva a számjegyeket egy meghatározott számot fejeznek ki. Pl. 2, 3, 4 egymás mellé írva 234. A tízes számrendszerben tíz különböző számjegyet használunk. Egy byte 256 különböző értéket vehet fel, azért a byte-ok egymás mellé írásával akár 256-os számrendszerben is dolgozhatunk. A carry flag és egy újabb összeadási utasítás jóvoltából ezt könnyedén meg lehet tenni.
Két szám összeadásakor a tízes számrendszerben azt kellett figyelnünk: eléri-e az azonos helyiértékű számjegyek összege a tízet – ha igen, átvitel keletkezett:

```
2   3   4
5   8   2
---------
8   1   6
1
átvitel
```

Visszatérve a több byte-on ábrázolt természetes számokra, ezek összeadásánál azt kell figyelni, hogy az azonos helyiértékű byte-ok összege eléri-e a 256-ot. A figyelést megoldja a C flag: jelzi, hogy van vagy nincs átvitel. Ezt az átvitelt azonban a következő helyiértéknél figyelembe kell vennünk. Erre egy újabb utasítás szolgál, az ADC (összeadás a carry-bit figyelembevételével). Jele
ADC A,s
ahol s az A, B, C, D, E, H, L, (HL), (IX+d), (IY+d) ill. n bármelyike lehet ugyanúgy, mint az ADD esetében. Az ADC utasítások kódjait a megfelelő ADD utasítás kódjából nyerhetjük, 8 hozzáadásával. Pl.:
ADD A, (HL) kódja 134,
ADC A, (HL) kódja 142
A flagek kezelése az ADD-éval azonos.
Ha a C átviteli flag értéke 0 az ADC utasítás végrehajtásának megkezdésekor, a végeredmény minden tekintetben azonos lesz, mintha ADD utasítást hajtottunk volna végre: az A regiszter tartalmához hozzáadódik az s regiszter, illetve memóriarekesz tartalma (ha s=A, akkor A tartalma megkétszereződik), s a végeredménynek megfelelően automatikusan beállítódnak a flagek. Maga az összeadás a 0–255 számábrázolásnak megfelelően történik, ezért a végeredmény 0-tól 510-ig adódik. A C flag mint 256 helyiértékű bit szerepel; ha az összeg legalább 256, a C flag 1 lesz, az A regiszter tartalma pedig az összegnél 256-tal kevesebb.
Ha a C átviteli flag értéke 1 az ADC utasítás végrehajtásának megkezdésekor, akkor az ADC utasítás hatása olyan, mintha egy ADD és egy INC A utasítást hajtottunk volna végre automatikusan egymás után, s a flagek állítása az INC utáni helyzetnek megfelelően történik.
A Carry (átvitel) flag figyelembevételét mutatják az alábbi példák:

Carry flag	0	0	1	1
A regiszter	7	7	7	7
C regiszter	15	251	15	251
ADD A,C	22	2	22	2
ADC A,C	22	2	23	3

Ne tévesszük el: az ADC kétszer is dolgozik a C átviteli flaggel: először is figyelembe veszik a Carry előző állapotát (0 vagy 1) mint összeadandót, az A regiszter új tartalmának kialakításához. Másodszor pedig a kialakult végeredménynek megfelelően állítja az F re-

PÁTRIA NYOMDA
NYOMTATVÁNYELLÁTÓ KIRENDELTSÉG



ÜGYVITELI MUNKÁJÁT MEGKÖNNYÍTI HA VÁLLALATUNK TERMÉKEIT HASZNÁLJA!

Nyomtatványellátó Kirendeltségünk ajánlata:

- Szabványos és szabványtól eltérő **ügyviteli nyomtatványok** hagyományos és önátírós kivitelben
- Számítógépekhez **leporellók** széles választékban
- Különböző **egyedi nyomtatványok** kigyártatása
- Műszaki célokat szolgáló **diagramok** és **regisztrálók**, lap és tekercs kivitelben, **telextekercsek**
- Különféle **jegyek** készíttetése kívánság szerint:
  - tombolák
  - ruhatárjegyek
  - belépők különböző rendezvényekre
  - kulturális rendezvények – színház, mozi, kiállítás, múzeum – belépőjegyek
  - sportrendezvények, uszodák belépőjegyei, bérletei
  - automatagépek tekercskivitelű jegyei

SZOLGÁLTATÁSAINK IGÉNYBEVÉTELÉVEL A KORSZERŰ ÜGYVITELI MUNKA ALAPJAIT BIZTOSÍTHATJA!

Szabványnyomtatványok és jegyek megrendelhetők:

Áruforgalmi osztály

Budapest V., Szent István tér 4. I. emelet
Tel.: 173-133, 173-188, 173-293

Levélcím: Budapest, Pf. 564. 1374

Egyedi nyomtatványok, diagramok megrendelhetők:

Vevőszolgálat

Budapest XIII., Radnóti M. u. 15/b. · Tel.: 400-202

giszter bitjeit, köztük a Carry flaget is: ha az összeg 256, vagy annál több, C flag=1, ha 0–255 közötti az eredmény, C flag=0.
Az ADD és ADC segítségével most már akár öt byte-os (vagy még hosszabb) természetes számokat is összeadhatunk.

Első összeadandó:	1. byte	2. byte	3. byte	4. byte	5. byte
	ADC	ADC	ADC	ADC	ADD
Második összeadandó:	1. byte	2. byte	3. byte	4. byte	5. byte

Az összeadást a legalacsonyabb helyiértékű byte-okkal kell kezdeni ADD utasítással (azért ADD, és nem ADC, hogy a C flag korábbi állapotát – amely most az összeadás szempontjából zavaró lenne – nehogy figyelembe vegye). Ezután helyiérték szerint növekvő irányba haladva sorra számítjuk a megfelelő helyiértékű byte-ok összegét – most már ADC-vel, mely az átviteleket folyamatosan figyelembe veszi. Természetesen csak (?!) olyan számokat adhatunk össze, amelyeknek összege is elfér 5 byte-on. A C flag ugyan jelzi, ha ennél nagyobb a végeredmény – de mit kezdjünk egy ilyen jelzéssel?
A gépi kódú összeadó program tényleges megírását Olvasóinkra hagyjuk. Javasoljuk, hogy először jelöljék ki az összeadandók, ill. az összeg helyét a memóriában: döntsék el, hogyan helyezik el helyiérték szerint a byte-okat: növekvő csak csökkenő sorrendben: HL, IX, ill. IY segítségével, s milyen megosztással jelölik ki az éppen összeadandó byte-okat. A kapcsoló BASIC programot azok is megírhatják, akik a gépi rutin megírásába még nem mernek belevágni: e BASIC program feladata lehet az adatok bekérése (INPUT), az ötbyte-os tárolás megoldása, a gépi rutin indítása, a végeredmény (az összeg és esetleg a flagek) kijelzése (PRINT)
A 8 utasítás hatását szemléltető pgrogram használatakor megfigyelhetjük, hogy összeadásnál a V túlcsordulásjelző (Overflow) flag akkor lesz 1, ha a végeredmény 128 vagy annál több. Szerepe tehát hasonló, mint a Carry flagé. Ez a V flag akkor jelentős, hogy egy byte-on +127, −128 közti előjeles egész számokat tárolunk, és ezekkel számolunk.

8 bites kivonás: SUB, SBC

Ez a két utasítás az ADD és ADC pontos megfelelője kivonás esetére. Itt az A regiszter tartalmából kivonhatjuk A, B, C, D, E, H, L, (HL), (IX+d) ill. nn bármelyikét. Az utasítás kódja a megfelelő ADD, ill. ADC utasítás kódjánál 16-tal több, pl.

SUB C kódja 145	ill. 91H
SBC A, (HL) kódja 158	ill. 9EH
SBC A, (IX+d) kódja 221 158 d	ill. DDH 9EH d

A flagek kezelése – az N kivételével – azonos az ADD-nál leírtakkal. Az N értéke SUB vagy SBC elvégzése után mindig 1. (Az N flaget a BCD, azaz a binárisan kódolt decimális számokkal végzett műveleteknél hasznosíthatjuk – de erről majd később.)

4. Összehasonlítás: CP

A CP s utasítás szintén A, B, C, D, E, H, L, (HL), (IX+d), (IY+d) ill. n bármelyikére vonatkozhat. A konkrét formák kódjai az ADD ugyanolyan formájú változatánál érvényes kódból kapható 56 hozzáadásával.
A CP s utasítás végrehajtásakor a mikroprocesszor ugyanúgy állítja be a flageket, mintha ugyanazokkal az adatokkal SUB utasítást hajtott volna végre – de sem az A, sem az s tartalma nem változik meg. Ennek megfelelően a C és a Z flag hordozza a „végeredményt”:

ha eredmény	A< s	A = s	A> s
C flag	1	0	0
Z flag	0	1	0

Ami a jelölést illeti, ha pl. s a konkrét esetben a (HL), akkor táblázatunkban A< (HL) azt az esetet jelzi, amikor az A regiszter tartalma kisebb, mint a (HL) címen álló memória-rekesz tartalma.
A Z, ill. C flag értékét a később sorrakerülő feltételes ugró utasításokkal és feltételes szubrutin hívásokkal vehetjük figyelembe.

5. Logikai műveletek: AND, OR, XOR

A 100. oldalon közölt programban még van három úgynevezett logikai művelet. Végrehajtásuk mindig az akkumulátor és egy másik egybyte-os adat között történik, a „végeredmény” az A-ba kerül, és a flagek is beállítódnak az eredménynek megfelelően. Mint a 190-es sornál látszik, a fenti műveletek kódja az ADD utasítás megfelelő alakjának kódjából 32, 48, ill. 40 hozzáadásával keletkezik (ld· a kódtáblázat is, 108. old.)
Az egyes műveletek jelentése: a két egybyte-os adat között bitenként kerül végrehajtásra az előírt logikai művelet. Átvitel egyik helyiértékről másikra nincs. A bitenkénti műveletvégzés szabálya:

egyik adat bitje	másik adat bitje	művelet	eredmény
0	0	AND	0
		XOR	0
		OR	0
0	1	AND	0
		XOR	1
		OR	1
1	0	AND	0
		XOR	1
		OR	1
1	1	AND	1
		XOR	0
		OR	1

A flagek kezelése:	C	Z	P/V	S	N	H
AND	0	↕	P	↕	0	1
OR	0	↕	P	↕	0	0
XOR	0	↕	P	↕	0	0

Itt új a P jelölés. Ez azt jelenti, hogy e három művelet esetén a P/V flag paritás flagként működik: értéke akkor 1, ha a végeredmény (A tartalma) páros darabszámú "1" bitet tartalmaz, és 0, ha páratlan darabszámút.
A többit jól bemutatja a közölt program. Javaslom, hogy az adatokat írjuk fel kettes számrendszerben, végezzük el az előírt műveleteket, és hasonlítsuk össze azzal, amit a program mutat. Figyeljük meg a flagek működését is! Néhány feladat elvégzése nem nagy idő, és megéri, mert BASIC-ben is hasznosítható fogásokat tanulunk belőle!

6. 16 bites összeadás és kivonás:

Kétbyte-os számok összeadása nem gond: arra kész gépi kódú utasítások vannak! Most az ADD utasításnak 3 alakja van:

ADD HL, ss	kódja 16∗ss+9
ADD IX, pp	kódja 221 16∗pp+9
ADD IY,rr	kódja 253 16∗rr+9

ahol ss a BC, DE, HL, SP jele 0, 1, 2, 3, kódolással, pp és rr lényegében ugyanez, de 2 az IX, illetve IY kódja lesz a második helyen is. Tehát ADD IX, IY nem létezik, de ADD IX, IX vagy ADD IY, IY igen. Ez az ADD utasítás sem veszi figyelembe a C flag korábbi állapotát, de a végeredménytől függően beállítja! Ezzel automatikusan folytathatjuk a több byte-os számok összeadását. Ezt még tovább könnyíti az

Utasítás:	Kódja:
ADC HL, ss	237 16∗ss+74 (EDH)

utasítás, (ss értékei, mint az előbb), amely figyelembe veszi a C flag előző állapotát. Kódja 237 16∗ss+74, tehát két byte az utasítás kódja is. Itt a végeredmény mindig a HL regiszterpárban keletkezik.
Az eddigieken kívül még egy 2 byte-os (16 bites) műveletet ismer a Z80: ez az

SBC HL, ss Kódja 237 16∗ss+66

szintén 2 byte a kód maga is, és a végeredmény kötelezően a HL-be kerül. Még érdekesebb, hogy nincs C flag nélküli változata. (Ez azt jelenti, hogy alkalom adtán a C flaget külön utasítással nulláznunk kell a kivonás megkezdése előtt.)
A felsorolt műveletek hatása a flagekre:

	C	Z	P/V	S	N	H
ADC A, s	↕	↕	V	↕	0	↕
ADD xx, yy	↕	.	.	.	0	?
ADC HL, ss	↕	↕	V	↕	0	?
SBC HL, ss	↕	↕	V	↕	1	?

A ? azt jelöli, hogy a szóbanforgó bit nem definiált.

Ha a program beválik,

akkor változtatni kell rajta!

(A programozás törvényeiből)

Ugró utasítások (elágazások). JP és JR

A különböző programozási nyelvekben – így a BASIC-ben is – nagyon fontos szerepük van a feltételes és feltétel nélküli ugró utasításoknak, szubrutinhívásoknak. Most ezek Z80 gépi kódú megfelelőiről lesz szó.

Feltétel nélküli ugró utasítások:

Utasítás	Kódja Tízes		Hexadecimális	
JR dis	24		18H	
JP nn	195		C3H	
JP (HL)	233		E9H	
JP (IX)	221	233	DDH	E9H
JP (IY)	253	233	DFH	E9H

Az utóbbi négy könnyen érthető: Jump=ugorj (ugyanaz, mint a GOTO) a megadott címre. A címet vagy magában a programban írjuk le; JP nn, vagy HL, IX, IY regiszterpár valamelyikében előállítottuk/tároltuk, és most már csak hivatkozunk rá. A JP nn forma tehát 3 byte helyet foglal el a memóriában: az utasítás kódját + 2 byte helyen az utasítás címét. Ne felejtsük el: először az alacsony, azután a magas helyiértékű byte kódolandó! Pl:

JP AE08H kódolása: C3, 08, AE ill. 195,8,174.

A JR dis forma működése egy kicsit összetettebb. A teljes utasítás (mind a két byte) beolvasása után a Z80 processzor a dis tartalmát hozzáadja a programlépés-számláló PC regiszterpár értékéhez (ugyanolyan értelmezéssel, ahogy azt a DJNZ dis kapcsán már részletesen leírtuk (l. a 96. oldalon)), és az így kapott címen folytatódik tovább a program. Így persze csak +129/–126 lépéssel távolodhatunk el a JR utasítás memóriabeli helyétől – de nem 3, hanem csak 2 byte szükséges a kódoláshoz. És ne felejtsük: sok kicsi sokra megy: ezer ugró utasításnál már csaknem 1 Kbyte a megtakarítás! Ami a végrehajtási sebességet illeti: kiemelkedően leggyorsabb a JP (HL), a többi ennek 2–2,5–3-szorosa.

Feltételes utasítások:

Utasítás	Kódja Tízes	Tizenhatos	Ugrás, ha..
JP NZ, nn	194	C2H	Z flag=0
JP Z, nn	202	CAH	Z falg=1
JP NC, nn	210	D2H	C flag=0
JP C, nn	218	DAH	C flag=1
JP PO, nn	226	E2H	P/V flag=0
JP PE, nn	234	EAH	P/V flag=1
JP P, nn	242	F2H	S flag=0
JP M, nn	250	FAH	S flag=1
JR C, dis	56	38H	C flag=1
JR NC, dis	48	30H	C flag=0
JR Z, dis	40	28H	Z flag=1
JR NZ, dis	32	20H	Z flag=0

Látható, hogy a feltétel csak 4 flag valamelyikének 0 vagy 1 értéke lehet. Ebből arra következtethetünk, hogy a BASIC interpreterek (a Z80 mikroprocesszoros gépeken) az IF utasításban leírható számtalan feltétel teljesülésétől, ill. nem teljesülésétől függően a fent említett flagek valamelyikét állítják be, majd azt tesztelik (vizsgálják).

A relatív ugrások (JR) kódja mindig 2 byte, (a második a már többször említett dis), a teljes alak viszont mindig 3 byte. Itt nincs JP (HL) típusú gyors és rövid változat.

Az ugró utasítások működése:

Az ugró utasítások – bármilyen meglepő első pillantásra – tulajdonképpen LD (Load!) típusú utasítások. Mint tudjuk, gépi kódban mindig az az utasítás következik, amelyiknek a címe az utasításkód beolvasásának megkezdése pillanatában a PC regiszterpárban található. Ezért úgy kell pl. a 40 000 címre ugrani, hogy PC-be 40 000-et töltünk. Ezt teszi a JP mindegyik változata, s – némi számolás után – a JR utasítások is. A feltételes utasításnál csak annyi az eltérés, hogy ha az utasításkóddal kifejezett feltétel nem teljesül, akkor a teljes utasítás (tehát mind a 2 vagy 3 byte beolvasása után) a PC tartalma változatlanul marad. Ez gyakorlat azt jelenti: ha a feltétel nem teljesül, a program az ugróutasítás után közvetlenül következő memóriacímtől, – ha viszont teljesül a feltétel, a kijelölt memóriacímtől folytatódik. A PC régi tartalmát sehol sem jegyzi fel a processzor, az elvész. Ezek szerint a JP és JR különböző fajtáit ugyanúgy használhatjuk, mint a GOTO és IF utasításokat BASIC-ben. A feltételek (a megfelelő flagek) beállítása a már részletezett ADD, ADC, SUB, SBC és INC, DEC utasításokkal történhet egyelőre (esetleg a szintén említett PUSH-POP technikával). Később még lesz szó olyan utasításokról, amelyek hatással vannak a flagek állapotára. Itt most csak annyit szögezzünk le: a JP és JR utasítások egyike sem állít flaget – habár a feltételes változatok felhasználják a flagek pillanatnyi értékét.

Szubrutin hívás

CALL és RETURN

Utasítás	Kódja tízes	tizenhatos	Feltétel
CALL nn	205	CDH	nincs
CALL NZ, nn	196	C4H	Z flag = 0
CALL Z, nn	204	CCH	Z flag = 1
CALL NC, nn	212	D4H	C flag = 0
CALL C, nn	220	DCH	C flag = 1
CALL PO, nn	228	E4H	P/V flag = 0
CALL PE, nn	236	ECH	P/V flag = 1
CALL P, nn	244	F4H	S flag = 0
CALL M, nn	252	FCH	S flag = 1

A felsorolt szubrutin hívások memóriaigénye 3–3 byte. A második és harmadik helyen – a JP-utasításokhoz hasonlóan – a kétbyte-os cím áll: először alacsony, azután a magas helyiértékű byte. A feltételek is ugyanazok lehetnek, mint a JP-nél. A nagyfokú hasonlóság magyarázata: a Call végrehajtása nagyon hasonlít a JP végrehajtásához, csak most a CPU feljegyezi a stackben, hogy honnan ugrik el. Tehát a CALL nn működése:

1. A Z80 beolvassa a Call-utasítás mindhárom byte-ját; a PC ekkor már a következő byte-ra mutat (az ábra a CALL 0C2FH utasítást mutatja, 0CH = 12; 2FH = 47):

CALL 205	n 47	n 12		

szubrutin kezdőcíme ↑ PC



KISVÁLLALATOK, SZÖVETKEZETEK, TÁRSULÁSOK

Nincs elég munkaerő? Elvégzik a munkát a gépek!
Dolgozzon mini vagy személyi számítógépekkel.
Az ezekhez szükséges programokat tőlünk készen megkaphatja!
A programokat az Ön kívánságaira, egyéni szükséglete szerintire alakítjuk, azok kezelésére a munkatársait betanítjuk és *rövid átfutási idő alatt az eredményt garantáljuk*

Gépek	*Programok*	
M08X	Főkönyvi könyvelés	(Főkönyv)
PROPER 8	Analitikus könyvelés	(Könyvel)
PROPER 16	Állóeszköz-nyilvántartás	(Á. eszköz)
COMMODORE 64	Költségfelosztás	(Üzi)
TAP 34	Alap- és segédanyag-nyilvántartás	(Anyag)
	Rendelésnyilvántartás	(Rendel)
valamint hasonló	Raktárnyilvántartás	(Raktár)
nagyságrendű	Munkaügy	(Munka)
mikrogépek	Bérelszámolás	(Bér)

Még egy szolgáltatásunkat felajánljuk: korlátozott mennyiségben *kisszámítógépeket bérbe adunk*, programokkal együtt!
Nem terheli a fejlesztési alapot, a bérleti díj „költségből" fizethető!!!

PRODUKTORG
SZERVEZÉSI VÁLLALAT

1251 Budapest, Fő u. 68.
Számítástechnikai Fejlesztési és Kisszámítógép Alkalmazási Iroda
Tel.: 350-169, 154-090/323 vagy 661 mellék.

2. A processzor a PC tartalmát elmenti a stackbe (tehát a SP tartalma is változik: kettővel csökken);
3. A processzor a PC-be tölti a programban szereplő szubrutin címet (az ábrán: OC2FH);
4. Folytatódik a program futása, a PC-ben levő címtől.
A feltételes CALL-utasításoknál a 2. és 3. lépés csak akkor hajtódik végre, ha a feltétel teljesül.
Látható, hogy a CALL nn a GOSUB ... pontos megfelelője. A feltételes szubrutin hívások BASIC párja: IF feltétel THEN GOSUB... Már ebből is sejthető, hogy a RETURN BASIC utasításnak is lesz gépi kódú párja. Van is, mégpedig:

Utasítás	Kódja tízes	tizenhatos	Feltétel
RET	201	C9H	nincs
RET NZ	192	C0H	Z flag = 0
RET Z	200	C8H	Z flag = 1
RET NC	208	D0H	C flag = 0
RET C	216	D8H	C flag = 1
RET PO	224	E0H	P/V flag = 0
RET PE	232	E8H	P/V flag = 1
RET P	240	F0H	S flag = 0
RET M	248	F8H	S flag = 1

Valamennyi utasítás egybyte-os. A feltétel nélküli RET utasítás a BASIC RETURN utasítás pontos megfelelője: a stackbe legutoljára eltett címen folytatódik a program futása. Ez a cím általában CALL utasítással került a stackbe, de nem mindig: egy PUSH utasítással eltett cím ugyanúgy megfelel. Persze, ha egy adatot felejtünk a stackben, a processzor azt is címként fogja értelmezni, ha egy RET utasítás végrehajtásakor találja meg!

Restart utasítások: RST

Ezek tulajdonképpen speciális szubrutin hívó utasítások: kódjuk egyben byte, és végrehajtási idejük is mindössze egyharmada a többi szubrutin hívásnak. Ennek ára: a szubrutin kezdőcíme kizárólag 0, 8, 16, 24, 32, 40, 48 vagy 56 lehet csak.

Utasítás	Kódja tízes		tizenhatos
RST p	199+p	ill.	C7+p

ahol p kizárólag az előbb felsorolt szubrutin címek valamelyike lehet.

Bit-műveletek: BIT, SET, RES

A Z80 CPU lehetővé teszi, hogy a hozzákapcsolt külső memória (ROM, RAM, ...) tetszőleges bitjének inverzét egyetlen műveleti lépésben a Z flagbe másoljuk. Ugyanez elérhető a CPU 7 db belső egybyte-os regiszterével kapcsolatban is:

Utasítás	Kód Decimális	Hexadecimális	Flagek C Z P/V S N H
BIT b,r	203 64+8∗b+r	CBH 01 ... ... (b, r)	. ‡ ? ? 0 1

Itt b értéke 0–7 lehet, b jelöli a 2^b helyiértékű bitet. Az r értéke 0–7 lehet, jelentése: B, C, D, E, H, L, (HL), illetve A. Pl. az E regiszter (jele 3) 32 helyiértékű, 5. bitje inverzének Z flagbe másolásakor 64+8·5+3 = 107,
tehát BIT 5, E kódja 203, 107, vagyis CBH 6BH.
Ha a HL regiszterpár tartalma 5E4CH, akkor a memória 5E4CH című byte-jának 2. (4 helyiértékű) bitje inverzének Z flagbe másolásakor 64+8·2+6 = 86, tehát
BIT 2, (HL) kódolása 203 86, ill. CBH 56H.
Mint látjuk, ezek az utasítások kétbyte-osak. Az utasítás kódjának fenti első felírásakor a „hexadecimális" oszlopban a második byte-ot bitenként (tehát kettes számrendszerben!) tüntettük fel. (Nem következetes, de talán jobban érthető.) (HL) helyett, szokás szerint, szerepelhet (IX+d), ill. (IY+d) is. Az átírási technika a szokásos, de mégis részletesen kiírjuk, mert a végeredmény szokatlan: négybyte-os utasításokról van szó (sok hűhó egy bitért?):
BIT b, (IX+d) kódolása decimálisan:
221 203 d 70+8·b
hexadecimálisan:
DDH CBH d 01...110 (... = b)

BIT b, (IY+d) kódolása decimálisan:
253 203 d 70+8·b
hexadecimálisan:
FDH CBH d 01...110 (... = b)

Vigyázzunk! a 221, 203 prefixek sorrendje kötött (és persze a többi byte sorrendje is kötelezően a feltüntetett!).
Gondolom, azt a Tisztelt Olvasó már régen kitalálta, hogy mire jó egy bit inverzének Z-be másolása! Természetesen: Z-től függő ugró (JP, JR, CALL, RET) utasításban hasznosíthatjuk. Vagyis további 1–2–3 byte-os utasításokkal elérhetjük, hogy a program futása a memória vagy az előírt regiszter tetszőleges bitjének értékétől függjön. Úgy is fogalmazhatunk, hogy a Z flag akkor lesz 1, ha a kérdezett (tesztelt) bit 0 – teljesen úgy, mint a korábbi utasításoknál: a Z flag azt jelzi, ha az eredmény nulla.
Ha a 64+8·b+r kódokat 128-cal megnöveljük (196+8·b+r), akkor a SET utasítást kapjuk, amely a memória, ill. az előírt regiszter megfelelő bitjét 1-re állítja. Ha viszont 64+8·b+r helyére 128+8·b+r kerül, a megfelelő bit törlődik (RES=reset), a bit 0 lesz.
A SET és RES utasítások nem változtatják meg a flagek állását.

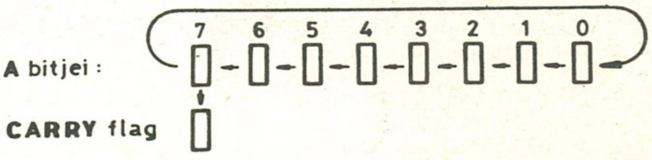
Rotálás, shiftelés:

Már korábban említettük, hogy a Z80 mikroprocesszor összeadáson (ADD, ADC) és kivonáson (SUB, SBC) kívül csak a „növeld meg eggyel!" (INC) és a „csökkentsd eggyel" (DEC) számtani műveleteket ismeri. Szorozni, osztani általában nem tud. Tud azonban 2-vel szorozni és osztani. Ez a kettes számrendszerben nagyon könnyű:

00101100 = 44 : 2 = 00010110 = 22

00101100 = 44 . 2 = 01011000 = 88

Látjuk: a számjegyeket egy helyiértékkel balra, illetve jobbra kellett eltolni. A szorzás/osztás szempontjából a legnagyobb/legkisebb helyiértékben álló „1" számjegy gondot okoz. Az ezekre vonatkozó döntéseinket be kell építenünk gépi kódú programjainkba.
A Z80 tervezői többféle, némileg eltérő jobbra/balra eltolási utasítást definiáltak. Ezek, aránylag természetes módon, három csoportba sorolhatók:
I. Az akkumulátorra vonatkozó egybyte-os utasítások:
RLCA (Rotode Left circular accumulator), amely balra tolja el az A akkumulátor bitjeit, az alábbi diagram szerint.



Az ábrából látható, hogy az A egyetlen bitje sem vész el: a 0–6. pozícióban levők eggyel magasabb helyiértékre kerülnek, az eredeti 7. bitről pedig két másolat is készül: az egyik a 0. pozícióba kerül, a másik a Carry flagbe. Ez az utasítás jól használható, ha egy byte minden egyes bitjére külön-külön van szükségünk. A vizsgálandó byte-ot betöltjük az A regiszterbe, majd egy RLCA következik. Ezzel az eredeti 7. helyiértékű bit a C flagbe kerül. Ezt egy C flag szerinti feltételes ugró vagy szubrutinhívó utasítással hasznosítva külön-külön leírhatjuk a 0, ill. 1 bit esetén szükséges programrészletet. Ezután visszaugorva az RLCA utasításra, feldolgozhatjuk a következő bitet. A nyolc bit feldolgozásáig (amit pl. egy DJNZ utasítással számolhatunk) az A regiszter minden bitje 8-szor tolódik el balra, így pontosan visszakerül eredeti helyére. Ezek szerint visszaállt az A regiszter eredeti tartalma, a vizsgálandó byte. Ha erre szükségünk van, külön memória- és időráfordítás nélkül rendelkezésünkre áll a programunk folytatásakor.
Az első csoportba még három utasítás tartozik:
RLA (Rotate left accumulator)
RRA (Rotate right accumulator)
RRCA (Rotate right circular accumulátor)

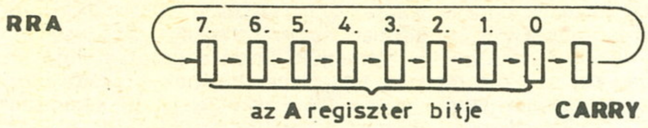
Bármely program,

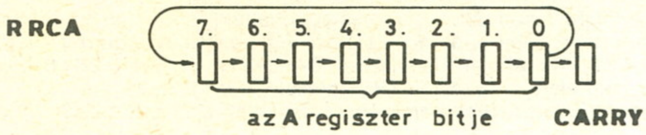
ha egyszer gépre kerül, elavult!

(A programozás törvényeiből)

Ezek diagramja:







Látható, hogy az RLA és az RRA utasításokban a Carry bit úgy viselkedik, mintha az A 8. bitje lenne. Az RLCA és az RRCA esetében a bitek körforgásában csak az A regiszter bitjei vesznek részt, a Carry bit pedig – a forgási iránytól függően – a legnagyobb, illetve legkisebb helyiértékű bit másolata lesz.
Kódolásuk:

Utasítás	Kód Tízes	Hexa	Flagek C	Z	P/V	S	N	H
RLCA	7	07H	↕	.	.	.	0	0
RLA	23	17H	↕	.	.	.	0	0
RRA	31	1FH	↕	.	.	.	0	0
RRCA	15	0FH	↕	.	.	.	0	0

Azt már az előzőekből is tudtuk, hogy a C flag (Carry-bit) értéket kap a művelet végzése során. A táblázatból az is látszik, hogy a Z (zero), P/V (paritás-túlcsodulás) és a S (signum-előjel) bit (flag) értéke változatlan marad, az N és a H bit pedig törlődik.
II. csoport: regiszterekre, memóriára vonatkozó eltolási műveletek: Az ide sorolt műveletek, a szokásos kódolás szerint, a B, C, D, E, H, L, A regiszterekre, illetve a HL regiszterpár által kijelölt memóriarekeszre vonatkozhatnak, s – ugyancsak szokás szerint – a (HL)-es változat (IX+d), illetve (IY+d)-re is működik, a megfelelő prefix használatával. A művelet végeredménye ugyanabba a regiszterbe, ill. memóriarekeszbe kerül, ahonnan a kiinduló adatot a processzor vette (tehát amit az utasításban az s helyére írunk). A flagek kezelése egységesen:

C	Z	P/V	S	N	H
↕	↕	P	↕	0	0

E műveletek mindegyikénél már a regiszterekre vonatkozó változat is két byte-os, az első byte 203(=CBH), a második a regisztertől és a művelettől függ:

Művelet:	Kódja:	Rövidítés magyarázata:
RLC s	203 s	Rotate left circular
RL s	203 16+s	Rotate left
RRC s	203 8+s	Rotate right circular
RR s	203 24+s	Rotate right
SLA s	203 32+s	Shift left
SRA s	203 40+s	Shift right
SRL s	203 56+s	Shift right logical

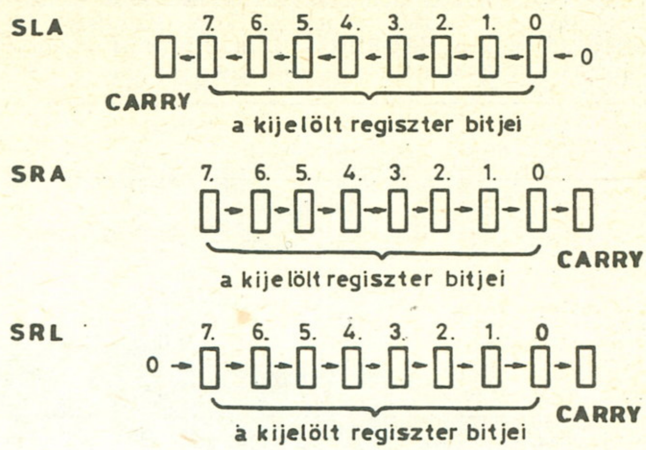
Emlékeztetőül: s „értéke":

regiszter	A	B	C	D	E	H	L	(HL), (IX+d), (IY+d)
s	7	0	'	2	3	4	5	6

Példaként néhány konkrét utasítás és kódolása:

RLC B	203 0	ill. CBH 00H
RL (HL)	203 22	ill. CBH 16H
RRC (IX+7)	221 203 7 14	ill. DDH CBH 07H 0EH
RR (IY−5)	253 203 251 30	ill. FDH CBH FBH 1EH

Látható, hogy az (IX+d), (IY+d)-re vonatkozó utasításoknál először jön az IX, ill. IY jele, a DD, ill. FD prefix, ezután a CB, ami szintén prefix jellegű, majd a d értéke, s végül a negyedik byte-ból derül ki, hogy melyik eltolási műveletről van szó.
Az első négy művelet ugyanúgy mozgatja a Carry flag és a kijelölt regiszter bitjeit, mint a hasonló nevű A-ra vonatkozó első csoportbeli utasítás. Ez azt jelenti, hogy pl. RLCA és RLC A (kódjaik: 7, illetve 203, 7) csak a Z, P/V és S flageket kezelik másként, egyébként ugyanaz a hatásuk (természetesen ez a pici eltérés pl. egy RET NZ, vagy JR Z, dis utasítás alkalmazásakor döntővé válik!).
A három shift (eltolás) művelet diagramja:

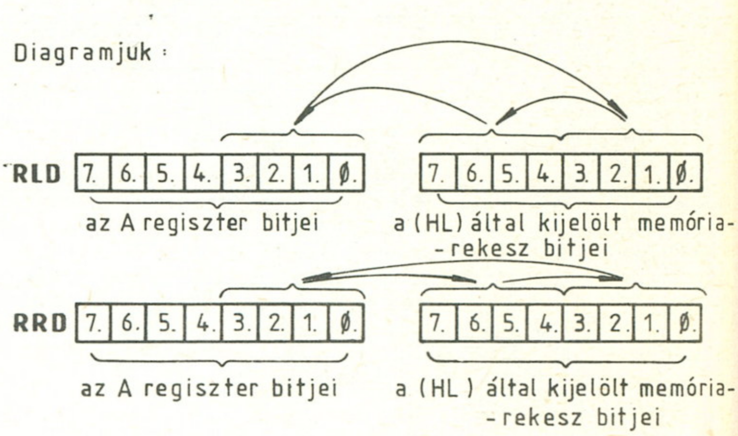


Ez az ábra is mutatja, hogy SLA tükörképe SRL, és nem SRA. Az SRA és SRL éppen a legnagyobb helyiértékű bit kezelésében tér el: SRL törli, SRA változatlanul hagyja azt.
III. csoport: félbyte-os mozgató utasítások:
Ebbe a csoportba mindössze két utasítás tartozik:

Utasítás				Kódolása	Rövidítés magyarázata:
RLD	237	111	ill.	EDH 6FH	Rotation left digit
RRD	237	103	ill.	EDH 67H	Rotation right digit

Az A akkumulátor és a (HL) által kijelölt memóriarekesz tartalmát mozgatják, 4–4 bitet együtt.

Diagramjuk:



Flagek kezelése:

C	Z	P/V	S	N	H
.	↕	P	↕	0	0

Ez hasonlít a második csoportéhoz, de a Carry bit nem változik meg. E két utasítás az úgynevezett BCD kódolású számábrázolás alkalmazása esetén tesz jó szolgálatot.

Blokkmozgatás, -keresés:

A következő 8 utasítással könnyen megoldható kisebb-nagyobb memóriaterületek átmásolása, vizsgálata:

Utasítás	Kódja Tízes		Tizenhatos		Flagek kezelése C	Z	P/V	S	N	H
LDI	237	160	EDH	A0H	.	.	↕	.	0	0
LDIR	237	176	EDH	B0H	.	.	0	.	0	0
LDD	237	168	EDH	A8H	.	.	↕	.	0	0
LDDR	237	184	EDH	B8H	.	.	0	.	0	0
CPI	237	161	EDH	A1H	.	↕	↕	↕	1	↕
CPIR	237	177	EDH	B1H	.	↕	↕	↕	1	↕
CPD	237	169	EDH	A9H	.	↕	↕	↕	1	↕
CPDR	237	185	EDH	B9H	.	↕	↕	↕	1	↕

Az utasítások jelentését egyenként, részletesen megadjuk:

LDI (Load, increment – betöltés növeléssel)

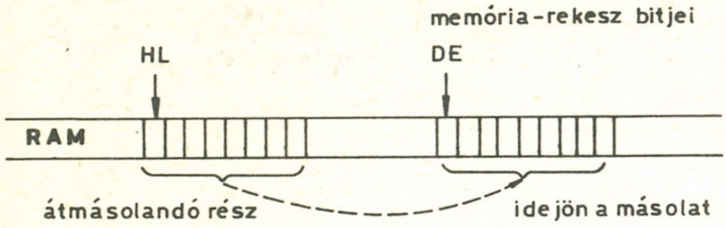
Az LDI utasítás végrehajtásakor a CPU megnézi, hogy mi a HL-regiszterpár és a DE regiszterpár pillanatnyi értéke, és a memória (HL) címen levő byte-ját átmásolja a (DE) címre. Ezután a HL és DE regiszterpár értékét eggyel növeli, a BC regiszterpár értékét eggyel csökkenti. Ha most a BC értéke 0, akkor a P/V flag is, egyébként ez a flag 1 lesz.

LDD (Load, decrement – betöltés csökkentéssel)

Az LDD utasítás az LDI párja: csaknem ugyanúgy működik, de most a HL és DE értékét nem növeli, hanem csökkenti eggyel a processzor. A BC csökkentése és a flagek kezelése is változatlan.

Hogy mi ennek az egésznek az értelme?

Ha egy ROM rutint némileg át akarunk alakítani, általában átmásoljuk a RAM-ba, s ott végezzük el az átalakítást. De ha a képernyő egy darabját mozgatni akarjuk, akkor is a memória egy részletét kell új helyre átmásolni:



Az eredeti helyen a memóriatartalom változatlanul megmarad. A másolás egy módja:

LD HL, honnan kezdődjön a másolás
LD DE, hová kerüljön a másolat
LD BC, hány byte-ot akarunk átmásolni
LDI
JP PE,...

Emlékeztetünk rá, hogy a JP PE,... utasítás azt jelenti, hogy csak akkor ugorjon a megadott címre, ha a P/V flag 1. Természetesen, a programba a ...helyére pontosan be kell írni az LDI utasítás memóriabeli kezdőcímét.

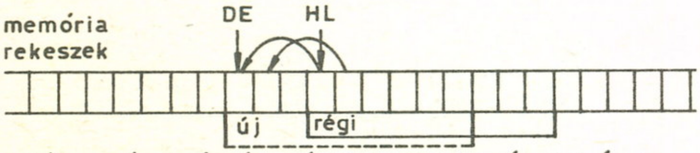
Már ez sem tűnik túl bonyolultnak: örülünk, hogy a HL és DE növelését az LDI automatikusan elvégzi. Így a következő LDI utasítás már a következő byte-ot másolja a következő címre. Sőt, a BC automatikus csökkentése miatt a P/V flag akkor lesz először 0, ha elfogytak az átmásolandó byte-ok.

Még ezt a programrészt is lehet egyszerűsíteni! Pontosan erre szolgál az LDIR utasítás: ez kerül majd az LDI és a JP PE,... helyére.

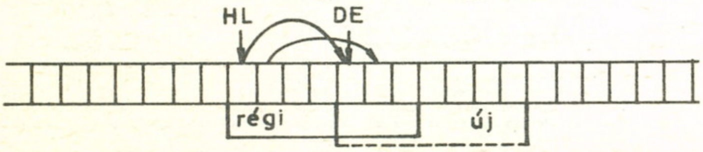
LDIR (Load, increment, repeat – töltsd, növeld, ismételd!)

Az LDI-nél leírt működést azzal kell kiegészíteni, hogy ha a BC tartalma nem lett 0, az LDIR az újonnan kialakult HL, DE, BC értékekre ismét végrehajt egy LDI utasítást. Ez azt jelenti, hogy mire az LDIR ciklus lejár, a teljes kijelölt memória rész átmásolódik, a BC értéke és a P/V flag biztosan 0 lesz. Ha az átmásolandó memóriaterület még részben sem fedi a kijelölt új memóriaterületet, az LDIR gond nélkül használható. A Z80-at nem érdekli, hogy DE tartalma kisebb vagy nagyobb, mint HL tartalma: vagyis előre-hátra könnyedén áthelyezhetjük a memóriarész tartalmát.

Ha azonban átfedés van a régi és új memóriaterület között, csak az egyik irányba problémamentes az átmásolás.

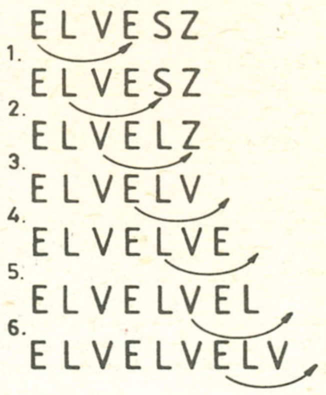


a hátra csúsztatásnál a régi tartalom egy része elvész:



A programozók legjobban
a káromkodás nyelvét ismerik!
(Troutman posztulátumaiból)

Ha ezt betűkkel, szöveggel szemléltetjük, talán még érthetőbb: 6 betű hátraírása:

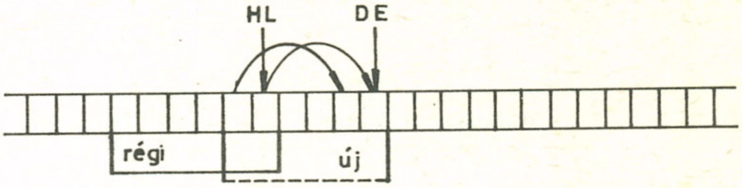


Ezért jó, hogy az LDIR-nek van egy párja, az

LDDR (Load, decrement, repeat – töltsd, csökkentsd, ismételd!)

Ez az utasítás az LDD-nél leírtakat ismétli, amíg csak a BC 0 nem lesz. Ekkor a P/V is 0, és pontosan BC memória-rekeszt másolt át.

Az LDDR utasításnál a régi és az új terület végét töltjük be a HL ill. DE regiszterpárokba:



Úgy kell választanunk az LDIR és LDDR között, hogy minden byte-ról előbb készüljön másolat, mint felülíródna. Ugyanezen elvek szerint választunk az LDI és LDD között. De mikor használjuk azokat, hiszen az LDIR és LDDR „összkomfortosabb"? Ha például a byte-okat nemcsak áthelyezni kell, hanem vizsgálni, esetleg egyes byte-okat kicserélni vagy beszúrni egyes jelek után/helyett egy byte-sorozatot, akkor az ismétlés nélküli utasítások lesznek segítségünkre.

CPI (Compare, Increment – összehasonlítás, növeléssel)

Az utasítás végrehajtása egy

CP (HL)
INC HL
DEC BC

utasítássorozatnak felel meg, eltérő flag kezelésse

1. összehasonlítja az A regiszter tartalmát a (HL) memória-rekeszben levő byte-tal. Ha egyenlőek, a Z flag 1 lesz, egyébként 0;
2. 1-gyel növeli HL értékét;
3. 1-gyel csökkenti BC értékét. Ha most BC=0 lett, akkor P/V értéke 0, egyébként 1 lesz.

CPD (Compare, decrement – összehasonlítás, csökkentéssel)

Minden ugyanúgy történik, mint CPI-nél, de most INC HL helyett DEC HL utasítást hajt végre a CPU, vagyis HL értékét eggyel csökkenti. A BC csökkentése és flagek állításának szabálya változatlanul megmarad.

CPIR (Compare, increment, repeat – összehasonlítás növeléssel, automatikusan ismételve)

A processzor először végrehajt egy CPI utasítást, majd megnézi, hogy a P/V flag=0 vagy Z flag=0 teljesül-e. Ezek bármelyikének (vagy mindkettőnek) teljesülése a CPIR végrehajtásának befejezését jelenti, egyébként újabb CPI következik, stb. Kevésbé technikai megfogalmazásban : a processzor a (HL) címen kezdve, a memóriában levő byte-okat (pontosabban: legfeljebb BC darabot) egyenként összehasonlítja az A regiszterrel. Ha egyezőt talál (Z flag=1), áttér a CPIR után következő utasítás végrehajtására. Ilyenkor BC tartalma alapján megállapítható, hogy (HL)-től kezdve hányadik byte egyezik A-val, HL alapján pedig azt, hogy ez hol van a memóriában.

Ha a memória tartalma (HL)-től kezdve BC rekeszen keresztül egyszer sem lesz A-val egyenlő, a processzor akkor is befejezi a vizsgálatot, de most a megálláskor BC=0, amit a P/V flag=0 is jelez.

Természetesen az is lehet, hogy pontosan a BC-edik (tehát az utolsó előírt) összehasonlítás járt eredménnyel, ilyenkor Z flag=1 és P/V=0 egyaránt teljesül.

Ezért CPIR után érdemes először a Z flaget vizsgálni (hogy volt-e egyezés, pl. JR Z, JP Z vagy CALL Z), majd a BC=0 feltételt tesztelni (P/V vizsgálata).

Ezek után természetes, hogy a

CPDR (Compare, decrement, repeat – összehasonlítás csökkentéssel, automatikusan ismételve)

utasítás mindössze abban tér el CPIR-től, hogy HL értékét nem növeli, hanem csökkenti eggyel. Az alkalmazás szempontjából ez azt jelenti, hogy CPIR elölről hátra, CPDR hátulról előre keres a memóriában.

Gyakorlásul írjunk programot, amely kikeresi az ERROR szót gépünk ROM-jából!

A Z80 és a külvilág kapcsolata

Input, output

Először is le szeretném szögezni, hogy a „külvilág" itt csak azt jelenti, hogy a CPU más áramkörökkel tart kapcsolatot. Ezek az áramkörök lehetnek a számítógép belsejében (többnyire így is van), de lehetnek a készüléken kívül is. A Z80 64K=65536 „külső" egységet, perifériát tud megkülönböztetni. Ezeket a címvezetékeken kiadott jellel különbözteti meg ugyanúgy, mint a memóriarekeszeket. Az írás és az olvasás is elég hasonló módon zajlik le elvileg, mint a memóriáknál. A memória, illetve periféria áramkörök úgy ismerik fel, hogy melyikükről van szó, hogy egy-egy speciális vezetéken ($\overline{MREQ}$ ill. $\overline{IORQ}$) a CPU ezt külön jelzi.

Utasítás	Kódolás Tízes	Hexa	Rekesz I/0	Periféria cím	C	Z	P/V	S	N	H
IN A, (n)	219 n	DB n	A	A n	.	.	.	.	.	.
IN r, (C)	237 64+8·r	ED ..	r	B C	.	↕	P	↕	0	↕
INI	237 162	ED A2	(HL)	B C	.	↕	?	?	1	?
INIR	237 178	ED B2	(HL)	B C	.	1	?	?	1	?
IND	237 170	ED AA	(HL)	B C	.	↕	?	?	1	?
INDR	237 186	ED BA	(HL)	B C	.	1	?	?	1	?
OUT (n), A	211 n	D3 n	A	A n	.	.	.	.	.	.
OUT (C), r	237 65+8·r	ED ..	r	B C	.	.	.	.	.	.
OUTI	237 163	ED A3	(HL)	B C	.	↕	?	?	1	?
OTIR	237 179	ED B3	(HL)	B C	.	1	?	?	1	?
OUTD	237 171	ED AB	(HL)	B C	.	↕	?	?	1	?
OTDR	237 187	ED BB	(HL)	B C	.	1	?	?	1	?

Az IN A, (n) utasítás végrehajtásakor a CPU a 256·A+n perifériáról olvas be. Ez azt jelenti, hogy az A regiszter tartalma kerül a címbusz felső nyolc bitjére, az utasításban szereplő n pedig a címbusz alsó nyolc bitjére. Ezután a CPU jelzi, hogy perifériáról akar olvasni, majd vár egy picit (ennek mértékét a hardver leírásból lehet ismerni), hogy a periféria megadhassa a válaszbyte-ot az adatbuszon, s az így kapott byte-ot elhelyezi az A regiszterben.

Az IN r, (C)utasítás végrehajtásakor az a különbség, hogy a BC regiszterpár tartalma lesz a periféria-cím. r értéke A, B, C, D, E, H, L és (HL) lehet, a szokásos kódolás szerint. Megemlítjük, hogy (HL) esetén a HL tartalma nem változik, csak a flageket állítja be a CPU. Ez jól használható akkor, ha egyszerre csak egy bitet viszünk át, mint pl. a ZX 81, Spectrum vagy HT1080Z kazettakezelése esetén.

Az OUT A, (n) és OUT (C), r utasítások végrehajtási logikája hasonló, csak a CPU nem beolvassa, hanem az adatbuszra kiküldi az adatot.

Ami az IN r, (C) és az OUT (C), r utasítást illeti, a jelölésben ugyan C szerepel, de a cím-buszon mindkét esetben BC jelenik meg. Minthogy a gyakorlatban egy byte is elég a perifériák megkülönböztetésére, a megfelelő logikai áramkörök gyakran csak az alsó byte-ot figyelik. Ilyen gépfelépítés esetén, a rendszer szempontjából a B regiszter tartalma közömbös.

A visszamaradó nyolc utasítás hasonló logikát követ, mint a LDI, ...LDDR utasítás négyese. Az előbbi táblázatból kiolvashatjuk, hogy a periféria-címet minden esetben BC adja, és a (HL) rekeszre vonatkozik az input, ill. output utasítás. A kiadott byte-okat most is BC számolja, a BC csökkentése és a HL változtatása automatikusan történik: HL tartalmát INI, INIR, OUTI és OTIR esetén növeli, a másik négy esetén csökkenti a CPU minden egyes utasítás végrehajtásakor. INIR, INDR, OTIR és OTDR az INI, IND, OUTI és OUTD automatikus ismétlését jelenti, amíg a B 0 nem lesz.

Mégegyszer hangsúlyozzuk, hogy ezen utasítás-csoport használatához részletesen kell ismernünk számítógépünk felépítését (pl. ZX 81 ill. Spectrum esetén az ULA működését).

Interrupt

Mint a bevezetőben már említettük, a Z80 CPU-nak két olyan kivezetése is van, amely működésének megszakítására szolgál. Ezek egyikének működése programból letiltható. A letiltható megszakítást maszkolható interruptnak, a másikat nem maszkolható interruptnak nevezzük. (Az előbbit szokás egyszerűen interruptnak nevezni.)

Ha a CPU nem maszkolható megszakítás kérést érzékel, akkor a folyó gépi kódú utasítás befejezése után a PC tartalmát elteszi a stackbe, a PC-be 0066H-ot tölt. Ez azt jelenti, hogy egy CALL 0066H utasítást hajt végre a megfelelő elektromos impulzus hatására. Pontosabban ilyenkor még az interrupt flip-flop tartalmát is beállítja. A 0066H címen kezdődően kell leírni azt a programrészletet (szubrutint), amit a nem-maszkolható megszakítás kérés érzékelésekor a CPU végrehajtani köteles. Ezt a szubrutint a RET utasítás egy módosított változata, a RETN zárja le. Ez az előbb említett flip-flopok helyes kezelését biztosítja. Végrehajtásakor a program futása ott folytatódik, ahol az interrupt kérés észlelésekor tartott.

Ha a CPU maszkolható megszakítás kérést kap, ezt csak akkor veszi figyelembe, ha ez engedélyezve van. A gép bekapcsolásakor a RESET hatására a maszkolható megszakítás-kérés mindig tiltott, annak figyelembevételét mindig programból kell engedélyezni EI (enable interrupt) utasítással. Ezt az engedélyt később bármikor visszavonhatjuk DI (disable interrupt) utasítással. Ha szükséges, akárhányszor ki-be kapcsolhatjuk a megszakítás-kérés engedélye-

	A	B	C	D	E	H	L	(HL)
ADD A,	87	80	81	82	83	84	85	86
ADC A,	8F	88	89	8A	8B	8C	8D	8E
AND	A7	A0	A1	A2	A3	A4	A5	A6
BIT 0,	CB47	CB40	CB41	CB42	CB43	CB44	CB45	CB46
BIT 1,	CB4F	CB48	CB49	CB4A	CB4B	CB4C	CB4D	CB4E
BIT 2,	CB57	CB50	CB51	CB52	CB53	CB54	CB55	CB56
BIT 3,	CB5F	CB58	CB59	CB5A	CB5B	CB5C	CB5D	CB5E
BIT 4,	CB67	CB60	CB61	CB62	CB63	CB64	CB65	CB66
BIT 5,	CB6F	CB68	CB69	CB6A	CB6B	CB6C	CB6D	CB6E
BIT 6,	CB77	CB70	CB71	CB72	CB73	CB74	CB75	CB76
BIT 7,	CB7F	CB78	CB79	CB7A	CB7B	CB7C	CB7D	CB7E
CP	BF	B8	B9	BA	BB	BC	BD	BE
DEC	3D	05	0D	15	1D	25	2D	35
INC	3C	04	0C	14	1C	24	2C	34
IN *,(C)	ED78	ED40	ED48	ED50	ED58	ED60	ED68	-
LD A,	7F	78	79	7A	7B	7C	7D	7E
LD B,	47	40	41	42	43	44	45	46
LD C,	4F	48	49	4A	4B	4C	4D	4E
LD D,	57	50	51	52	53	54	55	56
LD E,	5F	58	59	5A	5B	5C	5D	5E
LD H,	67	60	61	62	63	64	65	66
LD L,	6F	68	69	6A	6B	6C	6D	6E
LD (HL),	77	70	71	72	73	74	75	-
LD (IX+d),	DD77dd	DD70dd	DD71dd	DD72dd	DD73dd	DD74dd	DD75dd	-
LD (IY+d),	FD77dd	FD70dd	FD71dd	FD72dd	FD73dd	FD74dd	FD75dd	-
OR	B7	B0	B1	B2	B3	B4	B5	B6
OUT (C),	ED79	ED41	ED49	ED51	ED59	ED61	ED69	-
RES 0,	CB87	CB80	CB81	CB82	CB83	CB84	CB85	CB86
RES 1,	CB8F	CB88	CB89	CB8A	CB8B	CB8C	CB8D	CB8E
RES 2,	CB97	CB90	CB91	CB92	CB93	CB94	CB95	CB96
RES 3,	CB9F	CB98	CB99	CB9A	CB9B	CB9C	CB9D	CB9E
RES 4,	CBA7	CBA0	CBA1	CBA2	CBA3	CBA4	CBA5	CBA6
RES 5,	CBAF	CBA8	CBA9	CBAA	CBAB	CBAC	CBAD	CBAE
RES 6,	CBB7	CBB0	CBB1	CBB2	CBB3	CBB4	CBB5	CBB6
RES 7,	CBBF	CBB8	CBB9	CBBA	CBBB	CBBC	CBBD	CBBE
RL	CB17	CB10	CB11	CB12	CB13	CB14	CB15	CB16
RLC	CB07	CB00	CB01	CB02	CB03	CB04	CB05	CB06
RR	CB1F	CB18	CB19	CB1A	CB1B	CB1C	CB1D	CB1E
RRC	CB0F	CB08	CB09	CB0A	CB0B	CB0C	CB0D	CB0E
SBC A,	9F	98	99	9A	9B	9C	9D	9E
SET 0,	CBC7	CBC0	CBC1	CBC2	CBC3	CBC4	CBC5	CBC6
SET 1,	CBCF	CBC8	CBC9	CBCA	CBCB	CBCC	CBCD	CBCE
SET 2,	CBD7	CBD0	CBD1	CBD2	CBD3	CBD4	CBD5	CBD6
SET 3,	CBDF	CBD8	CBD9	CBDA	CBDB	CBDC	CBDD	CBDE
SET 4,	CBE7	CBE0	CBE1	CBE2	CBE3	CBE4	CBE5	CBE6
SET 5,	CBEF	CBE8	CBE9	CBEA	CBEB	CBEC	CBED	CBEE
SET 6,	CBF7	CBF0	CBF1	CBF2	CBF3	CBF4	CBF5	CBF6
SET 7,	CBFF	CBF8	CBF9	CBFA	CBFB	CBFC	CBFD	CBFE
SLA	CB27	CB20	CB21	CB22	CB23	CB24	CB25	CB26
SRA	CB2F	CB28	CB29	CB2A	CB2B	CB2C	CB2D	CB2E
SRL	CB3F	CB38	CB39	CB3A	CB3B	CB3C	CB3D	CB3E
SUB	97	90	91	92	93	94	95	96
XOR	AF	A8	A9	AA	AB	AC	AD	AE

	AF	BC	DE	HL	SP	IX	IY
ADD HL,	-	09	19	29	39	-	-
ADD IX,	-	DD09	DD19	-	DD39	DD29	-
ADD IY,	-	FD09	FD19	-	FD39	-	FD29
ADC HL,	-	ED4A	ED5A	ED6A	ED7A	-	-
SBC HL,	-	ED42	ED52	ED62	ED72	-	-
DEC	-	0B	1B	2B	3B	DD2B	FD2B
INC	-	03	13	23	33	DD23	FD23
LD rp,nn	-	01nnnn	11nnnn	21nnnn	31nnnn	DD21nnnn	FD21nnnn
POP	F1	C1	D1	E1	-	DDE1	FDE1
PUSH	F5	C5	D5	E5	-	DDE5	FDE5
LD rp,(nn)	-	ED4Bnnnn	ED5Bnnnn	2Annnn	ED7Bnnnn	DD2Annnn	FD2Annnn
LD (nn),rp	-	ED43nnnn	ED53nnnn	22nnnn	ED73nnnn	DD22nnnn	FD22nnnn

	-	Z	NZ	C	NC	M	P	PE	PO
CALL nn	CDnnnn	CCnnnn	C4nnnn	DCnnnn	D4nnnn	FCnnnn	F4nnnn	ECnnnn	E4nnnn
JP nn	C3nnnn	CAnnnn	C2nnnn	DAnnnn	D2nnnn	FAnnnn	F2nnnn	EAnnnn	E2nnnn
JR n	18nn	28nn	20nn	38nn	30nn	-	-	-	-
RET	C9	C8	C0	D8	D0	F8	F0	E8	E0

	0	1	2	3	4	5	6	7	
RST	C7	CF	D7	DF	E7	EF	F7	FF	restart utasítások

	növ.	növ+ism.	csökk.	csökk.+ism.	
CP	EDA1	EDB1	EDA9	EDB9	blokkban keresés
LD	EDA0	EDB0	EDA8	EDB8	blokk másolás
OUT	EDA3	EDB3	EDAB	EDBB	blokk output
IN	EDA2	EDB2	EDAA	EDBA	blokk input

	I	R	(nn)	(BC)	(DE)	(HL)	
LD A,*	ED57	ED5F	3Annnn	0A	1A	7E	A-ba másolás
LD *,A	ED47	ED4F	32nnnn	02	12	77	A-bol másolás

zését. Azt, hogy engedélyezett interrupt-jel érkezésekor mi történjen, programból befolyásolhatjuk. A CPU ugyanis három különbözőképpen képes az interrupt kezelésére. Ezeket az IM0, IM1, IM2 utasításokkal választhatjuk ki.

IM0 (interrupt mode 0)

Ennél az üzemmódnál, a megszakítást kérő egység az adat-buszon tetszőleges Z80 utasítást adhat a processzornak.

IM1 (interrupt mode 1)

Ennél az üzemmódnál a megszakítás-kérés egy 0038H címre vonatkozó szubrutinhívással egyenértékű, a flip-flopok kezelésétől eltekintve. A szubrutin végét RETI jelzi.

IM2 (interrupt mode 2)

Ez az üzemmód teszi lehetővé a legváltozatosabb választevékenységeket. Előkészítésként az I regiszterbe kell töltenünk a kiválasztott szubrutin cím-táblázat kezdőcímének felső nyolc bitjét. A megszakítás-kérés elfogadásakor a periféria szolgáltatja a szubrutin-cím táblázat címének alsó nyolc bitjét (ez célszerűen páros szám legyen). Az így kialakuló címen kell megadni a szubrutin kezdő címét.

nn	(IX+d)	(IY+d)	C	Z	P/V	S	N	H	
C6nn	DD86dd	FD86dd	*	*	V	*	0	*	hozzáadás A-hoz
CEnn	DD8Edd	FD8Edd	*	*	V	*	0	*	hozzáadás A-hoz CARRY-vel
E6nn	DDA6dd	FDA6dd	0	*	P	*	0	1	AND művelet A-val
-	DDCBdd46	FDCBdd46	.	*	?	?	0	1	a 0. bit tesztelése
-	DDCBdd4E	FDCBdd4E	.	*	?	?	0	1	az 1. bit tesztelése
-	DDCBdd56	FDCBdd56	.	*	?	?	0	1	a 2. bit tesztelése
-	DDCBdd5E	FDCBdd5E	.	*	?	?	0	1	a 3. bit tesztelése
-	DDCBdd66	FDCBdd66	.	*	?	?	0	1	a 4. bit tesztelése
-	DDCBdd6E	FDCBdd6E	.	*	?	?	0	1	az 5. bit tesztelése
-	DDCBdd76	FDCBdd76	.	*	?	?	0	1	a 6. bit tesztelése
-	DDCBdd7E	FDCBdd7E	.	*	?	?	0	1	a 7. bit tesztelése
FEnn	DDBEdd	FDBEdd	*	*	V	*	1	*	összehasonlítás A-val
-	DD35dd	FD35dd	.	*	V	*	1	*	csökkentés 1-gyel
-	DD34dd	FD34dd	.	*	V	*	0	*	növelés 1-gyel
-	-	-	.	*	P	*	0	*	input adott regiszterbe
3Enn	DD7Edd	FD7Edd	.	.	.	.	.	.	átmásolás A-ba
06nn	DD46dd	FD46dd	.	.	.	.	.	.	átmásolás B-be
0Enn	DD4Edd	FD4Edd	.	.	.	.	.	.	átmásolás C-be
16nn	DD56dd	FD56dd	.	.	.	.	.	.	átmásolás D-be
1Enn	DD5Edd	FD5Edd	.	.	.	.	.	.	átmásolás E-be
26nn	DD66dd	FD66dd	.	.	.	.	.	.	átmásolás H-ba
2Enn	DD6Edd	FD6Edd	.	.	.	.	.	.	átmásolás L-be
36nn	-	-	.	.	.	.	.	.	átmásolás (HL)-be
DD36ddnn	-	-	.	.	.	.	.	.	átmásolás (IX+d)-be
FD36ddnn	-		.	.	.	.	.	.	átmásolás (IY+d)-be
F6nn	DDB6dd	FDB6dd	0	*	P	*	0	0	OR muvelet A-val
-	-	-	.	.	.	.	.	.	output adott regiszterből
-	DDCBdd86	FDCBdd86	.	.	.	.	.	.	a 0. bit törlése
-	DDCBdd8E	FDCBdd8E	.	.	.	.	.	.	az 1. bit törlése
-	DDCBdd96	FDCBdd96	.	.	.	.	.	.	a 2. bit törlése
-	DDCBdd9E	FDCBdd9E	.	.	.	.	.	.	a 3. bit törlése
-	DDCBddA6	FDCBddA6	.	.	.	.	.	.	a 4. bit törlése
-	DDCBddAE	FDCBddAE	.	.	.	.	.	.	az 5. bit törlése
-	DDCBddB6	FDCBddB6	.	.	.	.	.	.	a 6. bit törlése
-	DDCBddBE	FDCBddBE	.	.	.	.	.	.	a 7. bit törlése
-	DDCBdd16	FDCBdd16	*	*	P	*	0	0	A+CARRY rotálása balra
-	DDCBdd06	FDCBdd06	*	*	P	*	0	0	A rotálása balra
-	DDCBdd1E	FDCBdd1E	*	*	P	*	0	0	A+CARRY rotálása jobbra
-	DDCBdd0E	FDCBdd0E	*	*	P	*	0	0	A rotálása jobbra
DEnn	DD9Edd	FD9Edd	*	*	V	*	1	*	Kivonás A-ból - CARRY
-	DDCBddC6	FDCBddC6	.	.	.	.	.	.	a 0. bit feltöltése
-	DDCBddCE	FDCBddCE	.	.	.	.	.	.	az 1. bit feltöltése
-	DDCBddD6	FDCBddD6	.	.	.	.	.	.	a 2. bit feltöltése
-	DDCBddDE	FDCBddDE	.	.	.	.	.	.	a 3. bit feltöltése
-	DDCBddE6	FDCBddE6	:	.	.	.	.	.	a 4. bit feltöltése
-	DDCBddEE	FDCBddEE	.	.	.	.	.	.	az 5. bit feltöltése
-	DDCBddF6	FDCBddF6	.	.	.	.	.	.	a 6. bit feltöltése
-	DDCBddFE	FDCBddFE	.	.	.	.	.	.	a 7. bit feltöltése
-	DDCBdd26	FDCBdd26	*	*	P	*	0	0	shiftelés balra aritm.
-	DDCBdd2E	FDCBdd2E	*	*	P	*	0	0	shiftelés jobbra aritm.
-	DDCBdd3E	FDCBdd3E	*	*	P	*	0	0	shiftelés jobbra logik.
D6nn	DD96dd	FD36dd	*	*	V	*	1	*	Kivonás A-ból
EEnn	DDAEdd	FDAEdd	0	*	P	*	0	0	XOR művelet A-val

```
                               | CCF          3F     CARRY átállítása
                               | CPL          2F     A bitenkénti negálása
hozzáadás HL-hez               | DAA          27     A decimális kiigazítása
hozzáadás IX-hez               | DI           F3     m. interrupt letiltása
hozzáadás IY-hoz               | DJNZ d       10dd   B csökk.+ugrás. ha poz.
hozzáadás HL-hez +CARRY        | EI           FB     m. interrupt engedélyezése
Kivonás HL-ből -CARRY          | EX DE,HL     EB     DE - HL csere
csökkentés 1-gyel              | EX AF,AF'    08     AF - A'F' csere
növelés 1-gyel                 | EXX          D9     BC-BC';DE-DE';HL-HL' csere
nn betöltése reg.párba         | EX (SP),HL   E3     (SP) - HL csere
átmásolás a stack-be           | EX (SP),IX   DDE3   (SP) - IX csere
átmásolás a stack-ből          | EX (SP),IY   FDE3   (SP) - IX csere
átmásolás (nn)-ből rp-be       | HALT         76     interrupt jelig vár
átmásolás rp-ból (nn)-be       | IM0          ED46   interrupt mód 0 beállítása
-------------------------------| IM1          ED56   interrupt mód 1 beállítása
                               | IM2          ED5E   interrupt mód 2 beállítása
szubrutin hivas                | IN A,(n)     DBnn   input A-ba
ugras adott cimre              | JP (HL)      E9     HL másolása PC-be
relativ ugro utasitas          | JP (IX)      DDE9   IX másolása PC-be
return                         | JP (IY)      FDE9   IY másolása PC-be
-------------------------------| LD SP,IX     DDF9   IX másolása SP-be
J E L M A G Y A R A Z A T      | LD SP,IY     FDF9   IY másolása SP-be
                               | LD SP,HL     F9     HL másolása SP-be
C  carry flag                  | NEG          ED44   A = 256-A
Z  zero flag                   | NOP          00     "nincs művelet"
P/V paritas/tulcsordulas       | OUT (n),A    D3nn   output A-ból
S   signum flag                | RETI         ED4D   interrupt-rutinból return
N Kivonas jellegu muvelet      | RETN         ED45   NMI-rutinból return
H half carry flag              | RLA          17     A+CARRY rotálása balra
                               | RRA          1F     A+CARRY rotálása jobbra
1  a flag feltoltodik          | RLCA         07     A rotálása balra
0  a flag torlodik             | RRCA         0F     A rotálása jobbra
*  a flag beallitodik          | RLD          ED6F   A-(HL) félbájtos,balra
.  a flag valtozatlan          | RRD          ED67   A-(HL) félbájtos,jobbra
?  a flag "nem definialt"      | SCF          37     CARRY flag feltöltése
```

Utasítás	Kódja Tízes	Hexa
EI	251	FBH
DI	243	F3H
IM 0	237 79	EDH 46H
IM 1	237 86	EDH 56H
IM 2	237 94	EDH E5H
RETI	237 77	EDH 4DH
RETN	237 69	EDH 45H

Ezek az utasítások a flageket nem állítják.
Már korábban említettük, hogy a HALT utasítás hatására a CPU addig vár, amíg interrupt jelet nem kap.

Csereutasítások

A bevezetőben említettük, hogy a Z80 CPU-nak A', F',. . . H',L' jelű regiszterei is vannak. Ezek kizárólag a most következő csereutasításokkal érhetők el.
EX AF, AF': ez az utasítás az AF és az A'F' regiszterpárok tartalmát cseréli fel.
EXX: ez az utasítás a BC, DE, HL regiszterek tartalmát cseréli fel a B'C', D'E', H'L' regiszterek tartalmával. E két csereutasítás végrehajtási ideje nagyon rövid, ezért előszeretettel használják gyors interrupt szubrutinokban. További csereutasítások is vannak, ezeket is tartalmazza a következő táblázat.

Utasítások	Kódja Tízes	Hexa
EX DE, HL	235	EBH
EX AF, AF'	8	08H
EXX	217	D9H
EX (SP), HL	227	E3H
EX (SP), IX	221 227	DD E3H
EX (SP), IY	253 227	FD E3H

Ezek az utasítások az EX, AF, AF' kivételével a flageket nem változtatják meg.

Egyéb utasítások

DAA: az A akkumulátor tartalmát BCD formának megfelelően kiigazítja a H és N flagek értékének figyelembevételével.
CPL: (complement accumulator): az A regiszter bitjeit ellenkezőre váltja át.
NEG (negate accumulator): az A tartalmának kettes komplemensét képezi. Értéke eggyel több, mint CPL eredménye.
CCF (complement carry flag): a carry flaget ellenkezőjére váltja át.
SCF (set carry flag): 1-et rak a carry flagbe.
Megjegyezzük, hogy a carry flag törlését SCF után adott CCF utasítással vagy pl. OR A utasítással érhetjük el.
NOP (no operation): a CPU regiszterei, flagjei nem változnak, de „végrehajtása" ugyanannai időt igényel, mint pl. a LD r, 'r' utasításé. Időzítési feladatoknál vagy programjavításnál használjuk.

Utasítás	Kódja tízes	hexa	C	Z	P/V	S	N	H
DAA	39	27H	↕	↕	P	↕	.	↕
CPL	47	2FH	.	.	.	.	1	1
NEG	237 68	EDH 44H	↕	↕	V	↕	1	↕
CCF	63	3FH	↕	.	.	.	0	?
SCF	55	37H	1	.	.	.	0	0
NOP	0	00H	.	.	.	.	.	.

Végül hat LD utasítást közlünk, amelyek jelölésmódjuk miatt nem szerepeltek a megfelelő fejezetben. Azóta megismerkedtünk ezekkel a jelölésekkel, reméljük, hogy most már önmagukért beszélnek.

Utasítások	Kódja Tízes	Hexa
LD (IX+d), r	221 112+r d	DDH 70+r d
LD (IX+d), n	221 54 d n	DDH 36 d n
LD (IY+d), r	253 112+r d	FDH 70+r d
LD (IY+d), n	253 54 d n	FDH 36 d n
LD r, (IX+d)	221 70+8·r d	DDH 46+8·r d
LD r, (IY+d)	253 70+8·r d	FDH 46+8·r d

Ezek az utasítások sem állítják a flageket.

ZSEB SZÓTÁR

AD konverter (vagy analóg digitál átalakító): folytonos elektromos jeleket kódokká alakító készülék.
AI (mesterséges intelligencia) gép: a számítógépek következő nemzedéke.
alfanumerikus karakterek: betűk, számok.
analóg digitál átalakító: folytonos elektromos jeleket időben és mértékben meghatározott egységekre bontó készülék.
ASSEMBLER: a gépi kódhoz közel álló számítógépes nyelv
beolvasási biztonság: hogy milyen eséllyel sikerül a kazettára kivitt programot a memóriába visszajátszani
berendezés-orientált áramkör: speciális feladatra (berendezéshez) készített egyedi (vagy kis szériás) áramkör
beszédszintetizátor: olyan szerkentyű, ami a gép mondanivalóját „emberi hangon" közli, felolvassa
bit: egy kettes számrendszerbeli helyiérték (0 vagy 1)
byte: 8 bitből álló memória-„egység"
botkormány: lásd joystick
bővítés: kiegészítő hardver csatlakoztatása
busz-csatlakozó: a központi egység adat- és címvonalaira való csatlakozás helye
cursor (ejtsd körzör): a következő nyomtatás vagy adatbevitel helye a képernyőn
DEBUGGER: program más programok „belövésére" azaz ellenőrzésére és a hibák javítására
DEF FN: függvényt definiáló BASIC-utasítás
DELETE (ejtsd dilit): sorok törlését lehetővé tévő parancs
DRAW (ejtsd dró): grafikai funkciók ellátásának megkönnyítésére szolgáló BASIC-utasítás
editálás: utasítások „átszerkesztése", törlése, kiegészítése, módosítása
elektroncső: hagyományos elektronikus jelerősítő alkatrész
ELSE (ejtsd elsz): az „IF" feltételes utasítás másik ága, az utána következő utasítás akkor hajtódik végre, ha a feltétel nem teljesül
EPROM: átprogramozható (égetéssel) csak olvasható memória
felbontóképesség: hogy hány sorban és oszlopban lehet pixeleket rajzolni
félgrafikus képernyő: alkalmi megfogalmazása annak, hogy a gép billentyűzete bizonyos grafikai jelek felvitelére alkalmas
fényceruza: olyan, a számítógéppel összekapcsolt „ceruza", amely a géphez kapcsolt megjelenítő adott pontjához érintve a pont koordinátáit a gép memóriájába továbbítja
fényesség: a Spectrumon a kiírt jelek és a háttér fényességének különbsége (kontrasztja) kétféle értékre állítható be
file kezelés: logikailag összetartozó adatok halmazának kezelése a számítógép által
FLASH (ejtsd fles): karakterek, feliratok villogtatására szolgáló utasítás. (A Spectrumnál minden kiírást is villogtathatunk ennek segítségével.)
floppy (disc): hajlékony lemez, amelyre mágneses elven adatokat rögzíthetünk
fordítóprogram: olyan program, amely a BASIC-programot lefordítja a gép saját nyelvére
frekvencia: hangmagasság
frekvenciatartomány: „hangterjedelem". (Utóbbi két szónak sok más értelmezése is lehetséges.)
félvezető: az integrált áramkörök alapanyaga többnyire szilícium vagy germánium
FULL SCREEN EDITOR: az egész képernyőn javíthatjuk vele programot tetszés szerint váltogatva a sorokat
gépi kód: a gép saját nyelve, a BASIC-utasításokat először erre fordítja le, csak azután tudja végrehajtani
grafikus karakterek: ezek a jelek, amelyekkel egyszerű ábrák rajzolhatók a képernyőre
hanggenerátor: a gép programozható „hangképző szerve"
IF: a BASIC-ben (és más nyelvekben) az egyik feltételes utasítás séma első szava (az angol szó jelentése: ha)
integrált áramkör: egyetlen (rendszerint félvezető) darabból, speciális technológiával kialakított, miniatürizált áramkör
intelligens terminál: önálló feladatok végzésre is alkalmas terminál
interface (ejtsd interfész): más gépekhez vagy perifériákhoz való kapcsolódási lehetőség
joystick (ejtsd dzsojsztik): mozgatható kar, amely jelet ad a gépnek, így például játékoknál figurák több irányú mozgatására alkalmas
karakter: a gép által megjeleníthető, előre rögzített jelkészlet valamelyik eleme
kimenő jelszint: a magnóból „kimenő" elektromos jel átlagos erőssége
karakterpozíció: egy „leütésnyi" hely a képernyőn (vagy sornyomtatón)
képdigitalizáló: az optikai jeleket bit-sorozattá alakítja át
katalógus áramkör: nagy sorozatban gyártott, szabványos, olcsó (integrált) áramkör
know-how: (ejtsd nóhau) technológiai, szervezési vagy más, a gyártásnál felhasználható ismeret
INKEY$: BASIC-függvény, értéke az éppen lenyomott billentyűnek megfelelő karakter
lapozás: olyan technika, amellyel választhatóan több (a HT-nél két „fél") képernyő jeleníthető meg
lemezmeghajtó (floppy drive): a számítógép mágneslemezes háttértárolásának műszaki eszköze
licence: engedély mások által kifejlesztett gyártási technológiák felhasználására
maszk: integrált áramkörök készítéséhez szükséges fotósabl
mátrixnyomtató: speciális nagy sebességű nyomtató
memória: adatok és programok tárolására szolgáló egység
memóriabővítés: a géphez csatlakoztatható külön memória (RAM)
MERGE (ejtsd mördzs): olyan utasítás, amelynek segítségével új programokat tudunk beolvasni a régi kitörlése nélkül
mikroprocesszor: (CHIP) a mikrogép „lelke", a gép működését vezérlő integrált áramkör
monitor: az az üzemmód, amelyben gépi kódú programokat lehet bevinni a gépbe és azokat ellenőrizni
nagy felbontású grafika: ha a gép a képernyőn sok pontot tud külön megjeleníteni
négyszögjel: állandó erősségű hangimpulzusok és szünetek váltakozása
nyomtató = printer: a számítógép által vezérelt „írógép"
ON: egy feltételes utasítástípus kulcsszava a BASIC-ban
ON ERROR GOTO: olyan BASIC-utasítás, amely a program futása közben előforduló hibák kivédésének segítését szolgálja
output port (ejtsd autput port): kimeneti csatorna, amelyen át különböző perifériák vezérelhetők
párhuzamos interface: egységeket összekapcsoló berendezés, ahol egyidőben sok információ oda-vissza továbbítására van lehetőség
periféria: a géphez csatlakoztatható megjelenítő, tároló és adatbeviteli eszközök
pixel: a grafikus utasítások által megjeleníthető legkisebb egység (pont vagy kis négyzet)
plotter: rajzgép
POKE: olyan BASIC-utasítás, amely lehetővé teszi, hogy a gép bármelyik byte-jának értéket adjunk
POP: utasítás, amellyel közvetlenül kivehetünk egy értéket egy speciális memóriából
printer: (lásd nyomtató)
PRINT AT: BASIC-ben kiírás a képernyő adott helyére.
PRINT USING (ejtsd print júzing): BASIC kiíró utasítás a kiírási formátum (például számjegyek száma, előjelek, kezdő 0-ák stb.) megadására

program: feladat végrehajtására összeállított utasítássorozat
RAM (angol betűszó): a gépet használó számára teljesen hozzáférhető (felülírható és kiolvasható) memóriaterület
RENUMBER (ejtsd rinámber): automatikus újrasorszámozás
RESET (ejtsd riszet): utasítás, amellyel a géphez kapcsolt megjelenítő bármely „felgyújtott" pontját kikapcsolhatjuk
reset (ejtsd riszet): –gomb segítségével a „megzavarodott" gép ismét alaphelyzetbe állítható
ROM: (angol betűszó) csak kiolvasható memóriaterület, amely a gép programozhatóságát biztosító „tudásanyagot" tartalmazza
RS 232: soros adatátvitelre szolgáló szabványos kimenet
SAVE (ejtsd széjv): olyan parancs, amellyel programokat és adatokat kazettán vagy mágneslemezen lehet tárolni
sebesség: adott idő alatt végrehajtott elemi utasítások száma
SCREEN $: (ejtsd szkrin): olyan függvény a Spectrumnál, amely megadja, hogy a képernyő adott helyén milyen karakter áll
SET: utasítás, amellyel a géphez kapcsolt megjelenítő bármely pontját „felgyújthatjuk"
SHIFT: olyasmi, mint az írógépen a betűváltó
soros interface: egységeket összekapcsoló berendezés, ahol csak egyféle információ egyidejű továbbítására van lehetőség
soros kapcsolás: nem párhuzamos, tehát nem teszi lehetővé a perifériák egyidejű működtetését
szilícium: félvezető elem
szintaxis: a programírásra vonatkozó formai szabályok összessége
szoftver: mindaz, ami a gépbe „beleírható"
TAB: BASIC-ben a kiírásnál előre megadható oszlopsorszám
tápegység: a hálózati feszültséget a berendezés szükségleteinek megfelelő mértékűre és típusúra átalakító berendezés
terminál: nagyobb számítógépek adatvégállomásai, innen lehet „kommunikálni" a géppel
TRACE (ejtsd trész): annyi mint nyomkövetés. Olyan parancs, amely egy program végrehajtása közben lehetővé teszi, hogy kövessük, hol tart épp a program.
tranzisztor: félvezetőből készült jelerősítő alkatrész
VERIFY (ejtsd verifáj): BASIC-parancs a kazettára vagy discre kivitt program ellenőrzésére, az eredetivel való összevetés révén

A SZUPER BIT-LET az ÖTLET számítástechnikai melléklete, a BIT-LET retrospektív különkiadása, amely a BIT-LET első 12 számára épül, de néhány későbbi anyagot is tartalmaz. • Szerkesztette: Angyalosi László • A szerkesztésben közreműködött: Halász Péter, Király Zoltán, Török Turul, Székely Jenő • Művészeti szerkesztő: Pribelszky Pál • Tördelő: Domokos Imre • Fotók: Gál Imre

SZUPER BIT-LET

BIT-LET-ünk jó hagyományai közé tartoznak a pályázatok. Volt már harmadgépnyerő, félgépnyerő, HT-nyerő, gépnyerő. Primo-nyerő, Micolor-nyerő, kétgépnyerő. Szuper BIT-LET-ünkből sem maradhat ki a pályázat tehát.
Ezúttal Lukács József *engedett szelíd rábeszélésünknek, s ajánlott föl pályázó kedvű olvasóinknak egyet a KIT-ként bevezetett HOMELAB 3 gépéből. Megkíméljük azonban a szerencsés nyertest az építkezés gyötrelmeitől, s Lukács József készen adja majd át a gépet.*
A feladat megoldásait 1986. május 31-ig lehet beküldeni az ÖTLET szerkesztőségének címére – Budapest 1986. Kérjük, hogy a Gépnyerő szelvényt a boríték külsejére szíveskedjenek fölragasztani!

Gépnyerő

HOMELAB 3

SZUPER BIT-LET

FELADAT:

Bőrönd Ödön és családja gyakran mennek nagyobb utazásra, s ilyenkor természetesen minden holmijukat bőröndökbe rakják, amikről azt kell tudni, hogy hatalmas nagyok, de elég gyengék; így 1–1 bőröndbe tetszőlegesen sok dolog kerülhet, ha az összsúlyuk nem több 6 kg-nál.
Ödön sohasem hajlandó részt venni a pakolásban. A felesége szívesen pakol, de csak akkor, ha egy héttel előbb megkapja a bepakolandó dolgok listáját (a súlyok feltüntetésével), ilyenkor azonban mindig képes minimális számú bőröndbe berakni az elvitelre szánt holmikat. (Itt és a továbbiakban feltesszük, hogy egyik holmi sem nehezebb 6 kg-nál.)
Ödön fia, Benedek, képes 2 óra alatt bepakolni, ha a holmik valahová le vannak rakva szépen sorban. A következő eljárást követi Benedek: sorba rakja az összes bőröndöt, aztán mindig megfogja a következő holmit, s megnézi, befér-e az első bőröndbe. Ha nem, akkor megy tovább, s végül az első olyan bőröndbe rakja, amelyikbe már belefér. Ha minden holmit elrakott, akkor az üresen maradt bőröndöket visszarakja a helyére. (Bőröndéknek rengeteg bőröndjük van, így mindig marad üres.)
Bőrönd Ödön lányának, Rebekának, egy délelőtt kell a bepakoláshoz; ugyanis ő először sorba rakja a holmikat, előre a legnehezebbet, s í. t., utoljára a legkönnyebbet. Majd ezután ugyanazt csinálja, amit az öccse, Benedek.
Ha adott egy L lista (azaz súlyok sorozata, mivel mint láttuk, a pakolásnál csak a holmik súlya lényeges), akkor ezt a „listát" a mama M(L), Benedek B(L), Rebeka R(L) db bőröndbe tudja elpakolni; világos, hogy ha ismerjük a listát, akkor M(L), B(L), és R(L) kiszámítható, s persze általában nem lesznek egyenlőek.

KÉRDÉSEK:

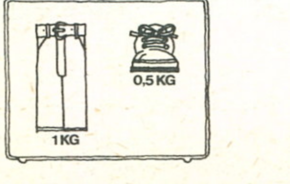
1. **Bizonyítsuk be, hogy akármilyen holmisorozat esetén Rebeka is és Benedek is kevesebb bőröndbe pakol, mint a mama által felhasznált bőröndök számának a kétszerese. Azaz lássuk be, hogy tetszőleges L lista esetén:**
$B(L) < 2 \cdot M(L)$; $R(L) < 2 \cdot M(L)$

2. **Adjuk meg holmiknak (ill. súlyoknak) egy olyan sorozatát, melyet a mama 3, Benedek és Rebeka viszont 4 bőröndbe tud bepakolni. Azaz egy olyan L1 listát kérünk, melyre:**
$M(L1) = 3$; $B(L1) = R(L1) = 4$

3. **Adjunk meg olyan holmi- (súly-) sorozatot, melyet a mama és Benedek 2, Rebeka viszont 3 bőröndbe pakol. Tehát olyan L2 lista kell, melyre:**
$M(L2) = B(L2) = 2$; $R(L2) = 3$

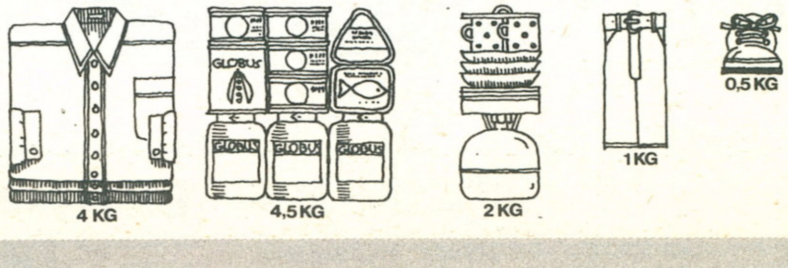
4. **Végül kérünk egy olyan holmi- (súly) sorozatot, amelyet a mama és Rebeka 6, Benedek viszont 10 bőröndbe pakol. Itt tehát olyan L3 listát kérünk, melyre:**
$M(L3) = R(L3) = 6$; $B(L3) = 10$

SZUPER BIT-LET GÉPNYERŐ



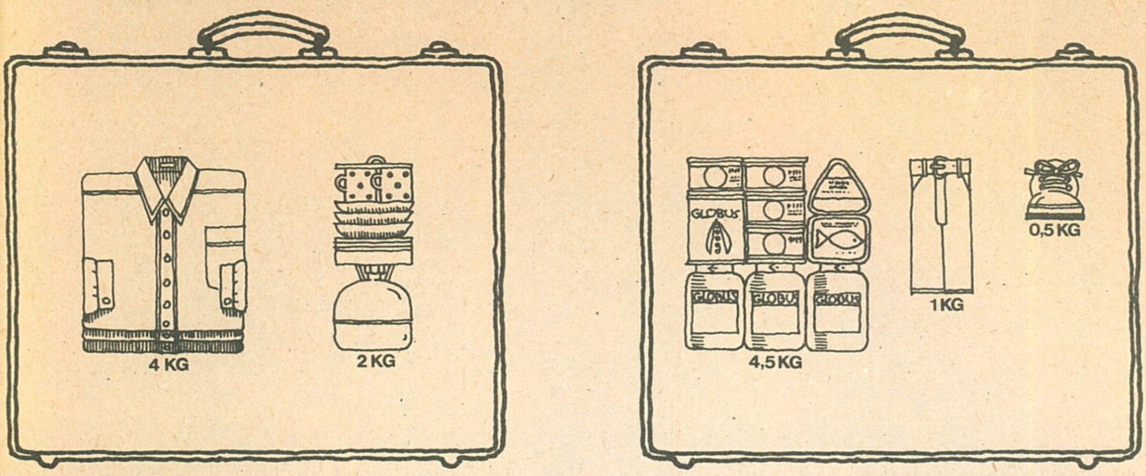
Ezt a szelvényt szíveskedjenek fölragasztani a megfejtést tartalmazó borítékra.
Beküldési határidő: 1986. május 31.

Bepakolandó holmik:
1–1 ilyen csomag előre elkészítve 1 egység
L = (4 kg; 4,5 kg; 5 kg; 1 kg; 2 kg)

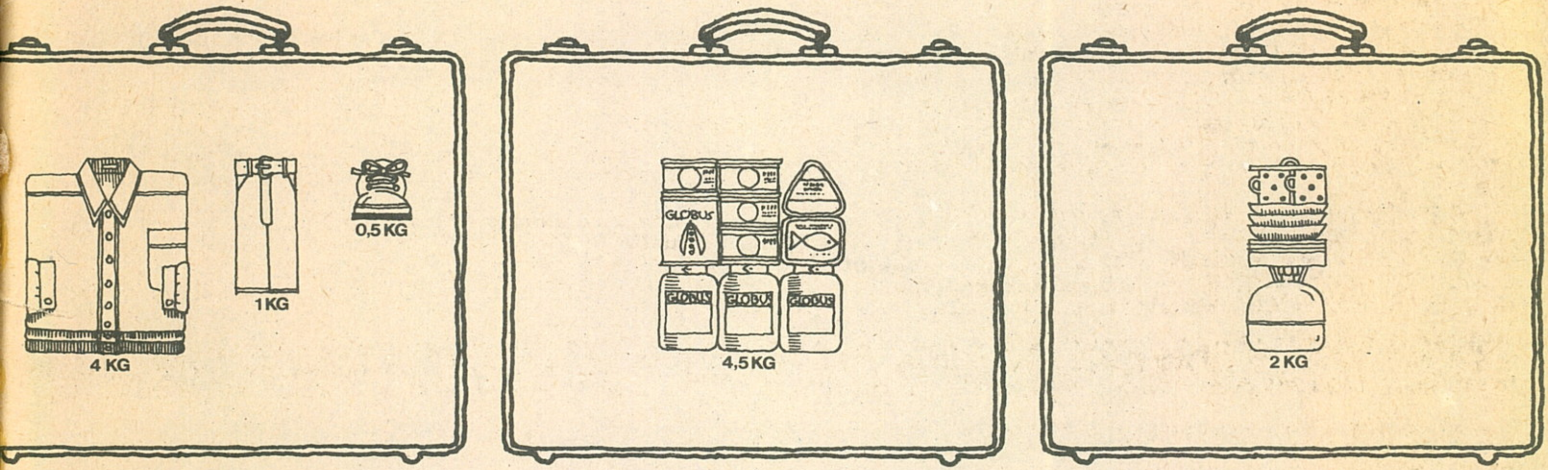


GÉPNYERŐ

A mama igy pakol:
tehát M(L) = 2



Benedek igy:
tehát B(1) = 3



Rebeka pedig igy:
Tehát R(L) = 2

